

TREBALL FI DE GRAU

Grau en Enginyeria de l'Energia

BIG DATA: DATA ANALYSIS AND DECISION MAKING



Memòria i Annexos

Autor: Marc Capdevila Prat
Director: Joan Martínez Sánchez
Convocatòria: Juny 2017

Resum

En aquest projecte es presenta un estudi del conjunt d'alternatives disponibles per a assistir el procediment de presa de decisions en l'empresa. L'estudi de les alternatives s'ha dimensionat per tal de treballar amb el màxim de camps possibles que s'han considerat relacionats amb la presa de decisions a l'empresa. Els camps per a l'estudi previ són: Teoria de decisions, "Business Intelligence" i "Sentiment Analysis".

Amb l'objectiu de desenvolupar una solució per a assistir el procés de decisió s'ha continuat amb disseny d'un sistema que realitza les tasques de Captura, Emmagatzematge, Anàlisi i Visualització dels resultats d'informació extreta a partir del Twitter. Aquest sistema està focalitzat a l'anàlisi de l'opinió expressada a través dels tweets.

Aquest sistema permet executar de forma independent qualsevol dels seus mòduls, o bé de captura o d'anàlisi i visualització. El mòdul de captura ens permet emmagatzemar tweets que compleixin certs criteris de filtratge i emmagatzemar-los en un fitxer de forma local. Els mòduls d'anàlisi i visualització analitzen les dades del fitxer i en mostren els resultats a través d'una interfície gràfica. Aquests mòduls ens permeten obtenir l'anàlisi de l'evolució de l'opinió, l'evolució de la mitjana d'opinió, la freqüència de graus d'opinió i el geoposicionament de tuits positius/negatius, entre altres anàlisis.

Resumen

En este proyecto se presenta un estudio del conjunto de alternativas disponibles para asistir el procedimiento de toma de decisiones en la empresa. El estudio de las alternativas se ha dimensionado para trabajar con el máximo de campos posibles que se han considerado relacionados con la toma de decisiones en la empresa. Los campos para el estudio previo son: Teoría de decisiones, "Business Intelligence" y "Sentiment Analysis".

Con el objetivo de desarrollar una solución para asistir el proceso de toma de decisiones se ha continuado con el diseño de un sistema que realiza las tareas de Captura, Almacenamiento, Análisis y Visualización de los resultados de información extraída a partir del Twitter. Este sistema está focalizado en el análisis de la opinión expresada a través de los tweets.

Este sistema permite ejecutar de forma independiente cualquiera de sus módulos, o bien de captura o de análisis y visualización. El módulo de captura nos permite almacenar tweets que cumplan ciertos criterios de búsqueda en un archivo de forma local. Los módulos de análisis y visualización analizan los datos del archivo y muestran los resultados a través de una interfaz gráfica. Estos módulos nos permiten obtener el análisis de la evolución de la opinión, la evolución de la media de opinión, la frecuencia de grados de opinión y el geoposicionamiento de tuits positivos/negativos, entre otros análisis.

Abstract

This project consists of a study of the alternatives available to assist the process of decision-making for businesses. The study of the alternatives has been dimensioned so that the maximum number of fields that were considered to be related to the decision-making procedure in companies can be studied. The fields chosen for the preliminary study are: Decision Theory, "Business Intelligence" and "Sentiment Analysis".

With the intention to develop a solution to assist the decision-making process, I have designed a system that performs the tasks of capture, storage, analysis and displaying of the results obtained through Twitter data. This system focuses on the analysis of opinion expressed through tweets.

The system modules can be executed independently, and perform the tasks of capture or analysis and visualization. The capture module allows us to store tweets that meet certain criteria and store them in a local file. Analysis and visualization modules analyze data from the file and display the results through a graphical interface. These modules perform different analysis such as evaluating the opinion's evolution, the evolution of the average opinion, the frequency and degree of opinion and geopositioning positive/negative tweets, among various other tasks.

Acknowledgements

Firstly, I would like to express my sincere gratitude to my tutor Joan Martínez Sánchez for all the support provided during the project, for his patience, motivation and knowledge. His guidance has been a key piece to the development of this project. I could have not imagined having a better mentor for my final project.

I would also like to thank my family: my parents and to my brother and sister for supporting and helping be become the person that I am today, and for giving me the opportunity to follow my own choices and aspirations.

Last but not least, I would love to thank Bianca for being there through the duration of the whole project and for listening when most needed it.



Glossary

- **Back-Office systems.** Set of technology services required to manage a company. These include systems for the IT, human resources and accounting departments.
- **Conditional Random Field (CRF).** A class of statistical modeling method often applied in pattern recognition and machine learning used for structured prediction.
- **Customer relationship management (CRM).** Strategy for managing all relationships and interactions with the customers and potential customers. Generally, it improves profitability.
- **Data Mining.** Computing process that aims to discover patterns in large data sets involving methods at the intersection of artificial intelligence, machine learning, statistics and database systems.
- **Enterprise resource planning (ERP).** Business process management software that allows an organization to use a system of integrated applications to manage the business and automate back-office functions.
- **Expectation maximization (EM).** Statistical iterative algorithm to find maximum likelihood or maximum a posteriori (MAP) estimates of parameters in statistical models.
- **Extract, transform and load (ETL).** Tools used to migrate data from one database to another, to form data marts and data warehouses and to convert data from one type to another.
- **Front-Office.** The main administrative office of a business or organization,
- **Hidden Markov Models (HMM).** A statistical Markov model in which a system with unobserved (hidden) states is modeled.
- **Latent Dirichlet allocation (MLSLDA).** A generative statistical model that allows a set of observations to be explained by observed groups that explain why some parts of the data is similar.
- **Machine learning.** Method of data analysis that automates analytical model building, by allowing computers to find hidden insights without being explicitly programmed.
- **Metadata.** A set of data that describes and gives information about other data.
- **Named entity recognition (NER).** Subtask of information extraction that aims to locate and classify named entities.
- **Natural language processing (NLP).** Field of computer science, artificial intelligence and computational linguistics concerned with the interactions between computers and natural languages.
- **One-vs-all (OVA) strategy.** Strategy that involves training a single classifier for each class.
- **Online analytical processing (OLAP).** An approach to answer multi-dimensional analytical queries in computing.
- **Online transaction processing (OLTP).** A class of information systems that facilitates and manage transaction-oriented applications.

- **Part of speech (POS).** A category in which a word is assigned in accordance with its syntactic functions.
- **Pointwise mutual information (PMI).** Measure of association used in information theory and statistics, referring to the average of all possible events.
- **Probabilistic latent semantic analysis (pLSA).** A statistical technique for analysis of two-mode and co-occurrence data.
- **Sales force automation (SFA).** Software that automates business tasks such as inventory control, sales processing and tracking of customer interactions, as well as analyzing sales forecasts and performances.
- **Sentiment Analysis.** Process of computationally identifying and categorizing opinions expressed in a text.
- **Singular value decomposition (SVD).** A factorization of a real or complex matrix.
- **Spectral feature alignment (SFA).** Algorithm that aligns domain-specific words from different domains into unified clusters.
- **Structural correspondence learning (SCL).** Method used for dealing with domain adaptation.
- **Structured Query Language (SQL).** It is a standard language for relational database management.
- **Supervised learning.** The Machine learning task of inferring a function from labeled training data.
- **Supply chain management (SCM).** Management of the flow of goods and services, tracking the movement and storage of raw materials, work-in-process inventory, and finished goods from point of origin to point of consumption.
- **Support Vector Machine (SVM).** A discriminative classifier formally defined by a separating hyperplane. Given a set of labeled training data, the algorithm outputs an optimal hyperplane which categorizes new examples.
- **Twitter API.** Service that provides access to read and write twitter data.
- **Unsupervised learning.** A type of machine learning algorithm used to draw interfaces from datasets consisting of input data without labeled responses.



Index

RESUM	I
RESUMEN	II
ABSTRACT	III
ACKNOWLEDGEMENTS	IV
GLOSSARY	VI
1. PREFACE	1
1.1. Motivation.....	1
1.2. Study of alternatives	1
1.3. Previous requirements.....	2
2. INTRODUCTION	3
2.1. Goal	3
2.2. Scope	3
3. DECISION THEORY	4
3.1. Decision Processes	4
3.1.1. Condorcet	4
3.1.2. Modern sequential models	5
3.1.3. Non-sequential models	5
3.2. Expected Utility	6
3.3. Bayesianism.....	7
3.4. Other variations of expected utility.....	8
3.4.1. Regret theory.....	8
3.4.2. Prospect Theory.....	8
3.5. Decision making under uncertainty.....	8
3.6. Decision making under ignorance	9
4. BUSINESS INTELLIGENCE	11
4.1. Architectures.....	13
4.2. Data marts and data warehouses.....	14
4.2.1. Characteristics	15

4.2.2.	Metadata.....	15
4.2.3.	Data	16
4.2.4.	Data structure	17
4.2.5.	Architecture	18
4.3.	Mathematical models.....	18
4.3.1.	Types of models	19
4.3.2.	Model development process	20
5.	BUSINESS INTELLIGENCE APPLICATIONS	22
5.1.	Relationship marketing.....	22
5.1.1.	Customer Relationship Management (CRM)	22
5.1.2.	CRM technological factors	23
5.1.3.	CRM implementation.....	24
5.2.	Sales force management	25
5.2.1.	Sales force automation (SFA).....	25
5.2.2.	SFA implementation	26
5.3.	Supply chain management	27
5.3.1.	SCM implementation	27
6.	SENTIMENT ANALYSIS	30
6.1.	Concept definitions.....	31
6.1.1.	Opinion definition	31
6.1.2.	Entity category and Entity expression	32
6.1.3.	Aspect category and Aspect expression	32
6.1.4.	Sentiment analysis tasks	33
6.1.5.	Types of opinions	33
6.1.6.	Subjectivity and emotion	34
6.2.	Document sentiment classification	35
6.2.1.	Supervised learning.....	35
6.2.2.	Unsupervised learning	37
6.2.3.	Sentiment rating prediction.....	38
6.2.4.	Cross-domain sentiment classification	39
6.2.5.	Cross-language sentiment classification.....	40
6.3.	Sentence sentiment classification	41
6.3.1.	Subjectivity classification	42
6.3.2.	Conditional and sarcastic sentences.....	43
6.4.	Aspect sentiment classification	43
6.4.1.	Aspect extraction	44

6.4.2.	Aspect sentiment classification	45
6.5.	Opinion summarization	46
6.6.	Opinion spam detection	46
6.6.1.	Individual vs group spamming	48
6.6.2.	Types of Data	48
6.6.3.	Supervised spam detection	48
6.6.4.	Unsupervised spam detection	49
6.6.5.	Group spam detection.....	50
7.	TWITTER SENTIMENT ANALYSIS	51
7.1.	Tasks and planification.....	51
7.1.1.	Field research	52
7.1.2.	Learning to program.....	52
7.1.3.	Development of the solution	53
7.2.	Tools	53
7.2.1.	Python.....	53
7.2.2.	Natural Language Toolkit	54
7.2.3.	Tweepy	55
7.2.4.	Other tools used	55
7.3.	Code	56
7.3.1.	Data capture	56
7.3.2.	Added accumulative sentiment	57
7.3.3.	Accumulative average sentiment.....	57
7.3.4.	Sentiment degree frequency.....	58
7.3.5.	Timely frequency	59
7.3.6.	Number of positive and negative tweets.....	59
7.3.7.	Number of tweets by country	60
7.3.8.	Number of positive/negative tweets by country.....	61
7.3.9.	Geolocation of positive/negative tweets.....	61
8.	RESULTS	63
8.1.	Added accumulative sentiment.....	63
8.2.	Accumulative average sentiment	64
8.3.	Sentiment degree frequency	65
8.4.	Timely frequency.....	66
8.5.	Number of positive and negative tweets	67
8.6.	Number of tweets by country	68

8.7. Number of positive/negative tweets by country.....	69
8.8. Geolocation of positive/negative tweets.....	70
CONCLUSIONS	71
Fulfillment of the original goal	71
Big Data as a tool for decision makers	71
ECONOMIC ANALYSIS AND COST EVALUATION	73
BIBLIOGRAPHY	77
ANNEX A. APPLICATION CODE	78
A1. Data Capture.....	78
A2. Added accumulative sentiment	79
A3. Accumulative average sentiment.....	80
A4. Sentiment degree frequency.....	81
A5. Timely frequency	82
A6. Number of positive and negative tweets.....	83
A7. Number of tweets by country	84
A8. Number of positive/negative tweets by country.....	85
A9. Geolocation of positive/negative tweets.....	86

1. Preface

The main objective of this project was to not only construct a working application involving twitter, sentiment analysis and graphs, but to learn about a new and fast growing field known as Big Data.

Big data is extremely valuable to increase productivity in businesses, but it also arises many challenges like difficulties in data capture, data storage, data analysis and data visualization. To define big data, the most commonly used expression is known as the 3Vs, which stand for *Volume*, *Velocity*, and *Variety*, indicating the size of the data sets, the speed in which the data is transferred and the different ranges of data types and sources.

Datasets are currently increasing at an exponential rate and because of this, there's a need for Big Data to stablish and make use of this fast-growing data.

1.1. Motivation

The first reason I wanted to make this project was because of my interest in programming, and specially in fields that extract data using data mining techniques and analyze that data hoping to find some interesting results. Despite being an energy engineer, I wanted to learn about something new that could actually be useful.

The second reason I wanted to specifically do a sentiment analysis on a stream of tweets was because on October 2016, I assisted on a Hackathon where I met other students who had done very similar projects, however, I wanted to make emphasis on the utility that this social media tool presents.

1.2. Study of alternatives

From the beginning, all I knew is that I wanted to make a project about how data analysis could assist the process of decision making for businesses. So, multiple alternatives were formulated besides the final choice:

- **Twitter sentiment analysis using Big Data specific tools.** This option was rejected because of the complexity. This solution would have required me to learn Scala, an object oriented scalable language aimed for large critical systems that work in clusters.
- **Text Mining over a set of unstructured data.** This approach consisted on manually collecting a set of unstructured text data. This data would then be used with some text mining software to obtain some results and trying to relate them to a decision-making process.

- **An overview of the common tools used in businesses to assist decision making through Big Data analysis.** From the beginning, this option seemed the most improvable because the vast majority of tools used by enterprises require an expensive monetary subscription

1.3. Previous requirements

Knowing that the chosen solution consists on developing an application that harvests tweets from twitter in real time, computes some sentiment analysis and proceeds to format the results into a visually understandable way. Therefore, the previous requirements for the chosen solution would be:

- Programming knowledge on the chosen language and libraries.
- Knowledge on common methodologies followed when making decisions.
- Expertise on field-specific terminology. Fields such as Big Data, Business Intelligence and Sentiment analysis.

2. Introduction

Due to the need for exploring new approaches to address the challenges of big data, companies are forced to forge their business modes accordingly. Big data is mainly used for analytic purposes and it enables the development of applications and real-time services that process massive amounts of data in order to present customers with value.

Big data can help customers receive a better service quality, help enterprises or small businesses predict market trends in order to adapt faster, plan their resources appropriately, etc. However, it also has its downsides like the need for a continuous development and maintenance of their infrastructures, and also involves certain costs that keep growing as the applications are being used.

2.1. Goal

The main Goal of the project is to present an easy introductory way to Big Data and some of its uses and applications. To do so, I plan on using tools that are easy to learn and programming languages that do not require much skill such as Python or Java.

2.2. Scope

This project aims to create a simple open-source based solution to perform sentiment analysis based on a dataset of twitter data, obtained thanks to the Twitter API. This solution would aim to perform all tasks any current Big Data application faces: Data Capture, Data Storage, Data Analysis and Data Visualization. However, on a much smaller scale.

The solution that is initially aimed for features a live tweet analysis, including live-updating graphs that allow to track the progress as it happens. However, this initial view of the project may be subject to change depending on the degree of difficulty that it presents.

3. Decision Theory

Decision Making is defined as the following: “The thought process of selecting a logical choice from the available options. When trying to make a good decision, a person must weigh the positives and negatives of each option, and consider all the alternatives. For effective decision making, a person must be able to forecast the outcome of each option as well, and based on all these items, determine which option is the best for that particular situation”. [1]

However, there’s an established theory involving all aspects of decision making known as “Decision Theory”.

Almost everything that a human being does involves decisions. Therefore, decision theory not only theorizes about decisions, but also about any human activity.

Decision theories can be either normative or descriptive, but the distinction about both theories is often blurred:

Normative theory is a theory about how decisions should be made as a previous step for rational decision-making. However, this rationality can also be complemented by ethical or political norms.

Descriptive theory it’s a theory about how decisions are actually made. As an example, this provides methods for a business executive to maximize profit.

3.1. Decision Processes

Decisions take time and it is natural to divide them into multiple steps or stages. There are multiple theories about the stages of a decision process such as **Condorcet**, **Sequential** and **Non-sequential** models.

3.1.1. Condorcet

The first generic theory about stages of a decision process was put forward by the philosopher Condorcet (1743-1794) as part of his motivation for the French constitution of 1793. He divided the decision process into three stages:

- 1) In this stage, one has to discuss the principles that will serve as the basis for the decision by examining various aspects of the issue and the consequences of the different ways to make the decision.

- 2) This second stage involves a second discussion in which the question is clarified, opinions are approached and combined with each other to a small number of more general opinions. In general terms, the decision is reducing to a choice between a manageable set of alternatives.
- 3) This third stage consists of the actual choice between the previous alternatives.

3.1.2. Modern sequential models

The modern sequential model has been evolving since John Dewey's first introduction. This original model consisted of five stages: A felt difficulty, the definition of the character of that difficulty, suggestion of possible solutions, evaluation of the suggestion and further observation and experimentation leading to acceptance or rejection of the suggestion

Herbert Simon modified Dewey's list to make it suitable for the context of decisions in organizations. This model consisted of three principal phases: Finding occasions for making a decision, finding possible courses of action and choosing among courses of action.

Further modifications were proposed by Brim and the process was divided into the following six sequential steps:

- 1) Identification of the problem.
- 2) Obtaining necessary information.
- 3) Production of possible solutions.
- 4) Evaluation of such solutions.
- 5) Selection of a strategy for performance.
- 6) Implementation of the decision.

3.1.3. Non-sequential models

The most influential model was proposed by Mintzberg, Raisinghani and Théorêt (1976). This vision stated that the decision process consists of distinct phases that do not have a sequential relationship. They used the same three phases as Simon, but gave them new names: Identification, development and selection.

- The identification phase consists of two routines: Decision recognition and diagnosis.
 - In the decision recognition phase, problems and opportunities are identified.
 - In the diagnosis phase, information channels are explored so that issues can be defined and clarified.
- The development phase serves to define and clarify the options. Consists of two routines:
 - The search routine aims at finding ready-made solutions.
 - The design routine aims at developing new solutions or modifying ready-made ones.

- The selection phase consists of three routines: Screen, evaluation-choice and authorization.
 - The screen routine acts as a filter so that suboptimal alternatives are eliminated.
 - The evaluation-choice routine is the actual choice between alternatives.
 - The authorization routine consists of the search for approval so that the solution is acquired.

The relationship between the previous phases is circular rather than linear and the decision maker may cycle within the Identification phase in order to recognize issues. The following diagram shows the relationship between phases and routines of the decision process.

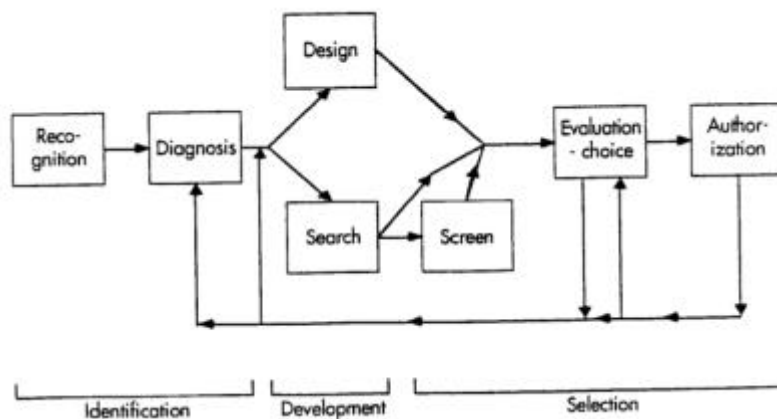


Figure 3.1. Relationship between phases and routines according to Mintzberg. Source: [2]

3.2. Expected Utility

The approach to decision-making that has had a major impact is known as Expected Utility (EU). It's considered to be the major paradigm in decision making for both descriptive and normative applications.

Expected utility is a “probability-weighted utility theory”. This means that to each alternative we assign a weighted average of its utility value under the different states of nature, the probabilities of these states are used as weights.

	State of nature 1 (p_1)	State of nature 2 (p_2)	...	State of nature n (p_n)
Choice 1	u_1	u_2	...	u_n

Choice 2	u_1	u_2	...	u_n
...
Choice 3	u_1	u_2	...	u_n

Table 3.1. Utility Table. Source: [2]

Given the previous table of utility values for each state of nature depending on the choice made. For each choice, we can calculate the expected utility by applying the following formula:

$$p_1 \cdot u_1 + p_2 \cdot u_2 + \dots + p_n \cdot u_n \quad (\text{Eq. 3.1})$$

In this case, given that the outcome is considered to be a better option depending on how high the utility value is, we would choose the option with the highest expected utility. This procedure is also known as “maximizing expected utility” (MEU).

3.3. Bayesianism

Expected utility theory with both subjective utilities and subjective probabilities is commonly called Bayesian decision theory or Bayesianism.

To summarize the ideas of Bayesianism, there are four principles, the first three refer to the subject as a bearer of a set of probabilistic beliefs, however, the fourth and last one refers to the subject as a decision-maker.

- 1) The Bayesian subject has a coherent set of probabilistic beliefs. This means that the subject follows a formal coherence or compliance with the mathematical laws of probability.
- 2) The Bayesian subject has a complete set of probabilistic beliefs. This means that the subject is able to assign a subjective probability. In addition, Bayesian decision-making is always considered to be decision-making under certainty or risk, never under uncertainty or ignorance.
- 3) When exposed to new evidence, the Bayesian subject changes his/her beliefs in accordance with his/her conditional probabilities.
- 4) Bayesianism states that the decision-maker chooses the option with the highest expected utility.

$$p(A | B) = p(A \& B) / p(B) \quad (\text{Eq. 3.2})$$

3.4. Other variations of expected utility

A great number of models for decision-making under risk have been developed from the EU theory.

3.4.1. Regret theory

In Expected Utility, an option is evaluated according to the utility that each outcome has irrespectively of what the other possible outcomes are.

Regret theory makes use of a two-attribute utility function that incorporates two measures of satisfaction: Utility of the outcomes (Same as in classical EU) and Quantity of regret. For each outcome, regret is measured as the difference in value between the possible outcome utility and the highest level of utility for all the other alternatives.

Therefore, regret theory is able to explain how a subject can either act by a risk prone or a risk averse behavior.

3.4.2. Prospect Theory

Developed by Kahneman and Tversky with the purpose to explain the results to an experiment with decision problem that were stated in terms of monetary outcomes and objective probabilities.

Prospect theory distinguishes two stages within a decision process:

- 1) **Editing phase.** The aim of this first phase is to organize and reformulate the options so as to simplify the following evaluation and choice. To achieve so, gains and losses are identified and defined relatively to a neutral reference point that usually corresponds to the current position.
- 2) **Evaluation phase.** During the evaluation process the decision-maker uses two criteria. A function that assigns a number to each outcome (Acts as some kind of subjective utility) and the objective probabilities for each “state of nature”.

3.5. Decision making under uncertainty

There are multiple decision criteria that aim to solve the problem that uncertainty presents:

- **Maximin Expected Utility (MMEU).** We should choose the alternative that its lowest possible EU is as high as possible. In simple words, we should maximize the minimal EU.

- **Reliability-weighted expected utility.** We should calculate the weighted average of probabilities, giving to each probability the weight corresponding to its degree of reliability. This criterion can be used to calculate a certain expected value for each alternative.
- **Ellsberg's index.** Daniel Ellsberg proposed an optimism-pessimism index that combines the previous two criteria.
- **Gärdenfors's and Sahlin's modified MMEU.** This decision rule makes use of a measure "p" or epistemic reliability that is chosen for each outcome. Then, a minimum level or epistemic reliability " p_0 " is established and choices with reliability lower than p_0 are excluded. After that, the Maximin Expected Utility criterion is used to the remaining possibilities.
- **Levi's lexicographical test.** Assuming that we have a permissible set of probability distributions and a permissible set of utility functions. A series of three lexicographically ordered tests or filters have to be applied.
 - E-admissibility. An option is E-admissible only if there is some permissible probability distribution and some permissible utility function such that they, in combination, make this option the best among all available options.
 - P-admissibility. An option is P-admissible if it is E-admissible and it is also best with respect to all other E-admissible options.
 - S-admissibility. For an option to be S-admissible it must be P-admissible and must have the highest minimum expected utility (MMEU) among all P-admissible alternatives.

3.6. Decision making under ignorance

Multiple decision rules are applied when it is known what the possibilities are, but no information about the probabilities is available:

- **Maximin rule.** For each alternative, we define its security level as the worst possible outcome with that alternative. Then, the maximin rule urges us to choose the alternative with the maximal security level. This principle was originally proposed by von Neumann as a strategy against an intelligent opponent. But this rule does not distinguish between alternatives with the same security level.
- **Leximin.** This rule distinguishes between alternatives with the same security level. It states that if two alternatives have the same security level, the one with the highest second-worst outcome is chosen. However, if both the worst and the second-worst outcomes are on the same level, then the third-worst outcomes are compared, etc.
- **Maximax rule.** We choose the alternative whose hope level (Best possible outcome) is best.
- **Optimism-pessimism index.** This rule lays between the maximin and the maximax rule. According to this rule, the decision maker has to choose an index between 0 and 1 that

reflects his degree of optimism or pessimism (α). For each alternative A, let $\min(A)$ be its security level and let $\max(A)$ be the hope level. Then the index is calculated according to the following formula:

$$\alpha \cdot \min(A) + (1 - \alpha) \cdot \max(A) \quad (\text{Eq. 3.3})$$

If $\alpha=1$, then this procedure is reduced to a maximin criterion and if $\alpha=0$, then it is reduced to a maximax criterion.

- **Maximax regret.** In order to apply this rule, a regret matrix must be created by assigning to each outcome the difference between the utility of the maximal outcome in its column and the utility of the outcome itself. Then the maximax regret criterion advises you to choose the option with the lowest maximal regret.
- In order to reduce ignorance to risk, the **principle of insufficient reason** can be used. This principle states that if there is no reason to believe that one event is more likely to occur than others, then the events should be assigned equal probabilities.

Decision value	Value information needed	Character of the rule
Maximin	Preferences	Pessimism
Leximin	Preferences	Pessimism
Maximax	Preferences	Optimism
Optimism-pessimism index	Utilities	Varies with index
Minimal regret	Utilities	Cautiousness
Insufficient reason	Utilities	Depends on partitioning

Table 3.2. Major decision rules for ignorance. Source: [2]

4. Business intelligence

The enormous data store technologies' cost reduction and the wide availability of internet connections have made it easier for individuals and organizations to generate and access enormous amounts of data.

This data contains commercial, financial and administrative transactions, web navigation paths, emails, texts, etc. This originated a question: Is it possible to convert this data into information and knowledge that can be used by decision makers to aid and improve the management of enterprises or the public administration?

And so, business intelligence was created and defined as a set of mathematical models and analysis methodologies that exploit the available data to generate information and knowledge useful for complex decision-making processes.

Decisions are constantly being made and can be more or less critical, have either long or short-term effects as well as involve people at various hierarchical levels. So, business intelligence provides tools and methodologies to make *effective* and *timely* decisions.

Effective decisions allow objectives to be reached in a more effective way, however, *timely* decisions are better suited for environments characterized by growing levels of competition and high dynamism. Therefore, allows enterprises to react rapidly to the actions of competitors and to adapt to new market conditions.

Any Business Intelligence analysis follows a cycle, but we should first be familiarized with the following concepts:

- **Data.** Represents a structured codification of either standalone content or content involving two or more entities.
- **Information.** Corresponds to the outcome of the extraction and processing activities carried out on data.
- **Knowledge.** Information is transformed into knowledge when it is used to make decisions and develop the corresponding actions. In simple words, information that has been used.

Now that these terms have been differentiated, the cycle of a business intelligence analysis is:

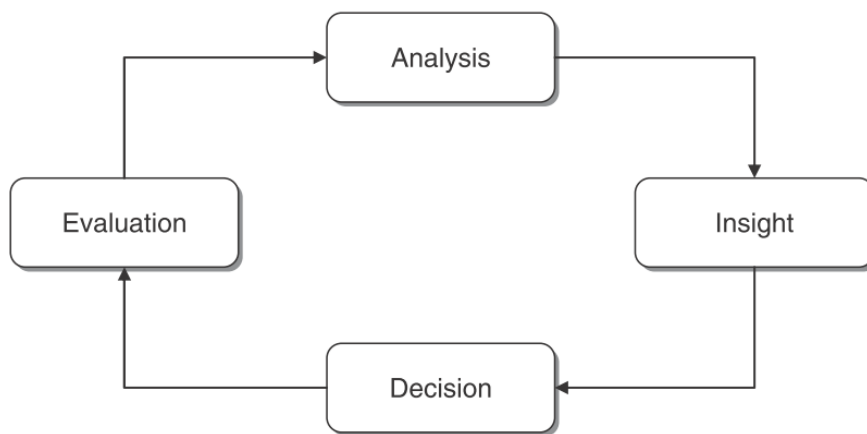


Figure 4.1. Cycle of a business intelligence analysis. Source: [3]

- **Analysis.** During this first phase, it is mandatory to recognize and describe the problem at hand. Decisions makers then create a mental representation of the phenomenon being analyzed, by identifying critical factors. This leads to decision makers asking several questions and obtaining quick responses in an interactive way.
- **Insight.** This phase allows decision makers to better and more deeply understand the problem itself. During this phase, the information obtained in the analysis phase is transformed into knowledge during the insight phase. This may occur due to the intuition of decision makers or through inductive learning models.
- **Decision.** Knowledge obtained as a result of the insight phase is converted into decisions and subsequently into actions.
- **Evaluation.** This fourth and last phase of the cycle involves performance measurements and evaluation. This evaluation is not limited to the financial aspects and should take into account the major performance indicators defined for the different company departments.

Even though the cycle is really simple, Business Analysis needs a diverse asset of tools that enable its implementation, technology, analytics and human resources.

- **Technology.** Hardware and software technologies are significant enabling factors. The exponential increase in microprocessor capabilities as well as the capacity of mass storage devices has enabled the use of advanced algorithms to be applicable to large sets of data, all within an affordable price.
- **Analytics.** Mathematical models and analytical methodologies play a key role in information enhancement and knowledge extraction from the available data, and even the visualization of the data plays a relevant role in order to facilitate the decision-making process. So, there is a

need to apply more advanced models of inductive learning and optimization in order to achieve forms of support for the decision-making process.

- **Human resources.** The human assets of an organization are essential to business intelligence. The ability of knowledge workers to acquire information and turning it into practical actions is the most important asset, as it has a major impact on the quality of the decision-making process.

4.1. Architectures

The architecture of a business intelligence system includes three major components:

- **Data source.** In first stages, it is necessary to gather and integrate the data stored in various sources. All this data is heterogeneous in origin and type.
- **Data warehouses and data marts.** By using extraction and transformation tools known as *extract, transform, load* (ETL), the data is stored in databases intended to support business intelligence analyses.
- **Business intelligence methodologies.** The extracted data is used to feed mathematical models and analysis methodologies in order to help decision makers.

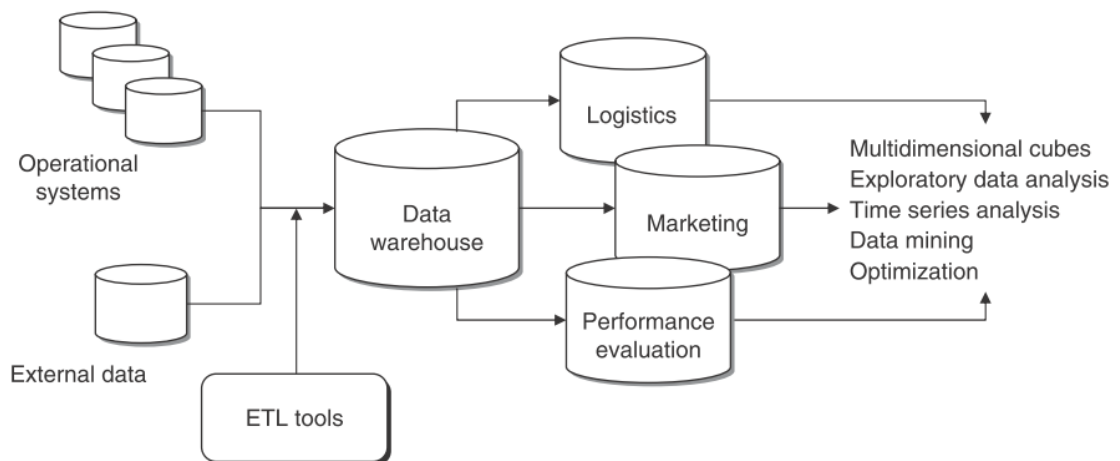


Figure 4.2. A typical business intelligence architecture. Source: [3]

Even though the typical architecture is integrated by the previous three main components, business intelligence methodologies include:

- **Data exploration.** Consists of a set of passive tools which consist of query, reporting systems and statistical methods. There are referred as passive tools because decision makers are requested to generate hypotheses and define the data extraction criteria.
- **Data mining.** Includes a set of active business intelligence methodologies which extract information and knowledge from data. These methodologies include mathematical models such as pattern recognition, machine learning and data mining technologies.
- **Optimization.** This model allows us to determine the best solution out of a set of alternatives.

4.2. Data marts and data warehouses

Data warehouse is defined as a massive database serving as a centralized repository of all data generated by all departments and units of a large organization. Advanced data mining software is required to extract meaningful information from a data warehouse. [4]

However, the term *data warehousing* includes a whole set of other activities such as design, implementation and usage of the data warehouse.

Data marts are systems that gather all the data required by a specific company department and can be considered a departmental data warehouse of a smaller size and a more specific type.

But before anything else, we should differentiate between *Online transaction processing* (OLTP) and *Online analytical processing* (OLAP).

- OLTP is characterized by a large number of online transactions. Its main objective is to be able to use very fast query processing, while maintaining data integrity and effectiveness. OLTPs' source of data is the original, as it records operational data. Its purpose is to control and run fundamental business tasks.
- OLAP is characterized by relatively low volume of transactions, but very complex queries with aggregations. These systems are used by data mining techniques and the data stored is consolidated data, which means that it comes from various OLTP databases, however, it also stores historic data. Its purpose is to help with planning, problem solving and decision making.

Despite the similarities between Online transaction processing and Online analytical processing, data warehouses for OLAP systems should not be compatible with OLTP applications. This will ensure the following:

- **Integration.** As a data warehouse contains heterogeneous information, it is required to facilitate access to the information by using uniform encoding methods. This would not be possible if the database had to be compatible with both OLTP and OLAP.

- **Quality.** Data must be examined and corrected to make sure that it's reliable and free of any error.
- **Efficiency.** Having databases compatible with multiple systems would risk the efficiency of the system by the need to use complex algorithms.
- **Extendibility.** Data from OLTP systems is usually removed after it is no longer valuable. Nevertheless, business intelligence systems need to access all available past data in order to be able to detect certain patterns.

4.2.1. Characteristics

Data warehouses that support decision support systems should have the following characteristics:

- **Entity oriented.** The data contained in the database should be primarily be concerned with the main entities of interest for the analysis, such as products, customers, orders and sales.
- **Integrated.** Data should be homogenized as it is loaded into the data warehouse.
- **Time-variant.** All data is labelled with the time and period to which it refers.
- **Persistent.** Data is usually not modified and held permanently. This makes it easier to manage read-only access for users.
- **Consolidated.** As the information comes from OLTP systems, it has already been consolidated and will be kept as it is.
- **Denormalized.** The warehouse is not structured in any normal form, therefore, there's a need for some redundancies, also known as metadata.

4.2.2. Metadata

Metadata is a specific information structure that describes the data contained in a data warehouse. For each attribute, it indicates the original source of the data, the meaning, and the transformations to which it has been subjected.

It is recommended that the metadata is kept updated and directly accessible to all the users. In particular, it should include the following information:

- Documentation about the data warehouse structure such as layout, logical views, dimensions, hierarchies, etc.
- Documentation about data genealogy.
- A list about the usage statistics of the warehouse.
- Documentation containing the general meaning of the warehouse.

4.2.3. Data

From all the data given to a data warehouse, three types can be identified:

- **Internal data.** It is usually gathered through transactional applications (OLTP) over the operations of the company. This data usually contains information about the main entities involved in the company processes like customers, employees, products, sales and suppliers. This data comes from three types of collection systems:
 - **Back-office systems.** Collect basic transactional records such as orders, invoices, production and logistics data.
 - **Front-office systems.** Collects data originated from call-center activities, customer support and the result of some marketing campaigns.
 - **Web-based systems.** These gather sales transactions on e-commerce website, visits and forms filled out by customers.
- **External data.** This data is collected by external agencies and include: Data relative to sales, market share, predictions for specific industries or even financial indicators. The external data also includes the information provided by *Geographic information systems* (GIS), applications that gather territorial data.
- **Personal data.** This data is collected by decision makers and it is meant to be for their personal use.

The need to verify, preserve and improve the quality of data is a constant concern of those responsible for the design and updating the data warehouse. The following major factors can affect data quality:

- **Accuracy.** In order to be useful for analysis, data has to be highly accurate. This is accomplished by checking that it can be represented correctly and values are within admissible ranges.
- **Completeness.** Data should not have large numbers or missing values. However, most machine learning and data mining techniques are able to minimize the effect of partial incompleteness.
- **Consistency.** Data must be consistent across the different data sources.
- **Timeliness.** Data should be frequently updated. It is recommended to update in a daily or at most weekly basis.
- **Non-redundancy.** Data repetition and redundancy should be avoided in order to avoid waste of memory and possible inconsistencies.
- **Relevance.** The data has to be relevant to the needs.
- **Interpretability.** The meaning of the data should be well understood and correctly interpreted by the analyst.
- **Accessibility.** Data must be easily accessible to analysts and decision support systems.

4.2.4. Data structure

The design of a data warehouse or a data mart is based on a multidimensional paradigm that ensures fast response times and revolves around two different types of data tables:

- **Dimension tables.** They correspond to primary entities that in most cases derive from master tables stored in OLTP systems. Each dimension is structured according to the hierarchical relationships. For example, the temporal dimension is usually based on the following two hierarchies: {day, week, year} and {day, month, quarter, year}.
- **Fact tables.** They usually refer to transactions and contain two types of data:
 - Links to dimensional tables required to reference information contained in each fact table.
 - Numerical values of the attribute that characterize the transactions.

However, depending on how the system is structured and how the references between data tables are made, we can find the following schemes:

- **Star schema.** The fact table is placed in the middle of the schema and is linked to the dimension tables through appropriate references.
- **Snowflake schema.** Includes dimensional tables connected to other dimensional tables.
- **Galaxy schema.** This scheme involves several fact tables interconnected with dimension tables, linked in their turn with other dimensional tables.

Even though these schemes simply are combination of different data tables, the representation can change depending on the number of dimensions represented.

For one or two dimensions, information can be easily represented with simple vectors or tables. For three-dimensional data structures, we need to use cubes in order to be able to represent. However, with fact tables linked to n dimension tables we obtain a lattice of cuboids, a type of aggregation equivalent to *Structured query language* (SQL).

Despite the complexity of the data, there are a set of tools that ease the task of data manipulation.

- **Roll up.** Navigation from detailed to aggregated information. It can be achieved either by proceeding upwards to a higher level along a single dimension or by reducing by one dimension.
- **Roll down.** Navigation from aggregated to more detailed information. Like the precious Roll up technique, it can be achieved either by moving through a single dimension hierarchy or by adding one dimension.

- **Slice and dice.** When manipulating data that has more than one dimension, the slice and dice technique allows us to focus on what is needed. It consists of two steps:
 - **Slice.** The value of an attribute is selected and fixed along one dimension.
 - **Dice.** A cube is obtained in a subspace by selected several dimensions simultaneously.
- **Pivot.** In order to obtain different views of the data cube, the axes can be rotated.

4.2.5. Architecture

Data warehouse systems should always include the following components:

- The data warehouse itself, along with the different data marts, contains the data and all the functions that allow the data to be accessed, visualized and modified.
- *Extract, transform and load* (ETL) tools, also known as *Back-end*. They allow data to be extracted, assembled and loaded into the warehouse.
- Business Intelligence and decision support systems. Represent the *Front-end* and allows the decision makers to carry out analyses and visualize the results.

A data warehouse can be implemented by following different design approaches:

- **Top-down.** Based on the overall design of the data warehouse. It follows a more systematic approach but it results in longer development times and higher risks.
- **Bottom-up.** Based on the usage of prototypes and system extensions. This approach is usually quicker and provides more tangible results, however, it lacks an overall vision of the whole system.
- **Mixed.** Based on the overall design of the data warehouse, but including a prototyping approach. This approach is highly practical and preferable over the others.

ETL refers to a set of software tools that are designed to perform three main functions:

- **Extraction.** Data is extracted from the available internal or external sources. The data can be selected depending on the information needed by decision support system.
- **Transformation.** The goal of the transformation is to improve the quality of the data extracted through the correction of inconsistencies, inaccuracies and missing values.
- **Loading.** Data is loaded into the tables to make it available to analysts.

4.3. Mathematical models

A model is a selective abstraction of a real system, designed to analyze and understand the behavior of a real system. According to their characteristics, models can be classified into:

- **Iconic.** An iconic model is a material representation of the real system whose behavior is imitated by replication.
- **Analogical.** It is also a material representation, but it imitates the real behavior by analogy rather than replication.
- **Symbolic.** An abstract representation of a real system. Its purpose is to describe the behavior of the system through a series of variables, parameters and relationships.

Depending on the probabilistic nature of a models, it can be classified as *stochastic* or *deterministic*.

- **Stochastic.** Some input information represents random events characterized by a probability distribution.
- **Deterministic.** All input data is supposed to be known with certainty.

Also, depending on the temporal dimension, a mathematical model can be either *static* or *dynamic*.

- **Static.** Considers a system and a decision-making process within one single temporal stage.
- **Dynamic.** Considers a system through different temporal stages, generally corresponding to a sequence of decisions.

4.3.1. Types of models

There are several types of mathematical models for decision making. The following models correspond to the main types used in business intelligence:

- **Predictive models.** They play a primary role in business intelligence systems as all departments of an enterprise make use of predictive information to make decisions. There are two main categories:
 - *Explanatory* models identify a relationship between the dependent variable and the set of independent attributes.
 - *Time series* models identify any temporal pattern.
- **Pattern recognition and learning models.** In a broad sense, they aim to understand the mechanisms that regulate the development of intelligence by extracting knowledge from past experiences and applying it in the future. These mathematical models are currently being used in fields such as image recognition, relational marketing and manufacturing process control. The aims of this models are to identify regular patterns and to help forecast the value that a random variable will assume in the future.

- **Optimization models.** Their primary objective is to identify the optimal decision according to a certain criterion: The final choice must have the minimum cost and the maximum payoff. These models are used in decision-making processes where a set of limited resources must be allocated in the most effective way. These models are currently used in logistics and production planning, financial planning, work shifts planning, price determination, etc. Also, the mathematical models used can be linear optimization, integer optimization, convex optimization, network optimization and multiple-objective optimization.
- **Project management models.** A project can be defined as a set of interrelated activities carried out in pursuit of a specific goal, therefore, projects require planning and process control. To achieve that, project management makes use of mathematical models, specially network models that represent all the component activities of a project and the precedence relationships among them. As a clear example of a project management model we have the *Project Evaluation and Review Techniques* (PERT).
- **Risk analysis models.** Used when the decision maker is required to choose among a number of alternatives while having uncertainty regarding the effects of these alternatives. These models are based on Bayesian and Utility Theories.
- **Waiting line models.** Their purpose is to study the congestion phenomenon occurring when the demand for a service and its provision differ. To do so, these models introduce a number of components:
 - The *population* as the source from which the potential customers are drawn and to which they return after the service has been delivered.
 - The *arrival process* describes how customers arrive to the entry point of the system.
 - The *service process* describes how providers meet the requests of the customers.
 - The *number of stations* as an additional parameter of the system.
 - The *Waiting line rules* describe the order in which customers are extracted from the line to be admitted for service.

4.3.2. Model development process

In order to develop a mathematical model for decision making, the process to follow can be divided into four stages:

- 1) **Problem identification.** The problem has to be identified so that the correct hypothesis can be formulated.
- 2) **Model formulation.** The formulation depends on a group of factors that have to be taken into account:
 - a) **Time horizon.** As models usually include a temporal dimension, the time span of a model and the length of the basic intervals have to be considered.

- b) **Evaluation criteria.** Consists of a number of measurable performance indicators. They will establish a criterion for the evaluation of the different alternative decisions.
 - c) **Decision variables.** Symbolic variables representing alternative decisions.
 - d) **Numerical parameters.** It is mandatory to identify and estimate all numerical parameters of a system.
 - e) **Mathematical relationships.** All the mathematical relationships of a system must be identified. They can be either *deterministic* or *probabilistic*.
- 3) **Development of algorithms.** The solution is chosen and implemented through software tools that incorporate it.
- 4) **Implementation and testing.** When the model is fully developed, it needs to be implemented taking into account the following factors:
- a) Plausibility and likelihood of the conclusions
 - b) Consistency of the results at extreme values.
 - c) Stability of the results when minor changes are introduced.

5. Business intelligence applications

5.1. Relationship marketing

Communication-based marketing models provide a type of direction for companies wanting to focus their efforts better in acquiring, retaining and growing relationships with customers and other stakeholders. There has been a gradual increase in the importance of communication in marketing, this is demonstrated by the ability to differentiate these new marketing approaches from the traditional ones.

These models have proven that there are common theoretical roots of communication theory and marketing theory that parallel and enrich each other, therefore marketing has become more dependent on communication and then the concept of Brand Communication was created.

Brand message is the information sent by anything that a company does, as a result, this message must be oriented at corporate, marketing and marketing communication levels. Strategic consistency is necessary so that the messages are more coherent and appropriate for the audiences.

Although production is responsible for the quality of the product delivered, marketing is responsible for the perception of its quality.

One of the main characteristics of Relationship marketing is that it focuses more on retaining customers rather than obtaining new ones as it has been proven that a 5 percent increase in customer retention results in an average increase of customer lifetime value of between 35 and 95 percent, leading to significant improvements in the company's profit.

The need for a set of tools to manage and monitor customer relationships made it possible for *Customer Relationship Management* (CRM) to be implemented. CRM is a combination of people, processes and technology that seeks to understand the company's customers.

5.1.1. Customer Relationship Management (CRM)

In the mid-twentieth century, mass production techniques and mass marketing changed the competitive landscape by increasing product availability for consumers. However, the purchasing process was fundamentally changed, as customers lost their uniqueness and became an "account number".

On the contrary, nowadays companies are trying to re-establish their connections to new as well as existing customers in order to increase long-term loyalty. This can be achieved by the implementation

of CRM applications. These applications link the front-office (sales, marketing, customer service, etc.) and the back-office (finances, operations, logistics, human resources, etc.) through the general touchpoints: Internet, e-mail, sales, direct mail, telemarketing operations, call centers, advertising, fax and stores.

CRM technologies are not merely applications for marketing, sales and service departments, but rather, when fully implemented, it becomes a cross-functional, customer-driven, technology-integrated business process that maximizes relationships and surrounds the entire organization.

Their main goal is to maximize the profitability of customer interactions by offering them customized offers, simplicity and convenience for completing transactions.

Some of the software solutions currently available are: Oracle, SAP, PeopleSoft, Clarify, SAS and Siebel. Most of these are vendors responsible for developing *Enterprise Resource Planning Systems* (ERP). However, despite the implementation of a CRM software, it does not guarantee an easy solution as there are still risks such as project failure, inadequate return investment, unplanned budget revisions, unhappy customers, loss of employee confidence and misuse or mismanage of time or resources. In most cases, failed CRM projects are the result of companies lacking a profound understanding on the purpose of CRM.

5.1.2. CRM technological factors

Information technologies have been long recognized to enable the radical redesign of business processes in order to achieve great improvements in performance for organizations. This can be achieved by creating innovative methods to link companies with their customers, suppliers and stakeholders to take full advantage by analyzing data to find patterns, behaviors, models and even manage personalized experiences.

However, the true objective of a CRM system is to accumulate, store, maintain and distribute customer knowledge throughout the organization.

For a CRM system to work properly, it needs a *Data Warehouse*, an *Enterprise Resource Planning (ERP) system* and an *Internet connection*.

- The *Data Warehouse* would allow instant access to information by collecting and creating a historical record of all customer's interactions through consolidation, correlation and transformation of data. Data warehousing provides benefits such as accurate and faster access to information, data quality and filtering to eliminate bad and repeated data and data analysis tools to consolidate and evaluate data.

- *Enterprise resource planning systems*, when successfully implemented, link all areas of a company into a tightly integrated system with shared data and visibility.
- The massive growth that *The Internet* has experienced these past years has brought a new meaning to customer relationships by the introduction of e-customers and services delivered in ways that are traditionally impossible.

5.1.3. CRM implementation

For a CRM system to be implemented successfully, an organization has to shift its focus from a mass market and mass production to a more personalized way to fit its customers' requirements and market pressures. This would require firms to narrow their market segments, but would provide a higher customer fidelity, which would result in a customer lifetime value increase, leading to a significant improvement in profitability.

The statement "Retaining customers is more profitable than building new relationships" is true in the existing internet market. The *Boston Consulting Group* estimated that it costs \$6.80 to market existing customers via web, versus \$34 to acquire new web customers. [5]

So, it is a continuous effort that requires redesigning core business processes starting from the customer feedback. The following list corresponds to a series of steps defined for designing a customer-centric organization:

- 1) Ease the customer experience.
- 2) Focus on the end customer.
- 3) Redesign the front office and examine all the information that flows between the front and the back office.
- 4) Encourage customer loyalty by interacting with them.
- 5) Build measures to check and evaluate continuously in order to keep improving.

However, an Organization also requires a change to its organizational culture as it is the individual employees who are building relationships with the customers.

The most important change is the commitment coming from top managerial positions, it is needed much more than the blessing coming from the CEO, managers need to support and commit to CRM during the whole implementation.

Without this support and commitment, momentum dies quickly and the managers should set CRM initiatives for leadership, strategic direction and in alignment with the vision and business goals.

CRM also requires a full-time attention from the implementation team. This team should have representatives from sales, marketing, manufacturing, customer services, and Information technology.

When it comes to the rest of the organization, employees should get used to sharing information and knowledge enterprise-wide. These changes can be aided by implementing communication systems that reach all levels of employees.

5.2. Sales force management

The current need for sales force management systems has been due to an important trend in business-to-business marketing. This trend has shifted firms towards enhancing customer relationships and productivity through the deployment of Customer Relationship Management (CRM) and Sales Force Automation (SFA) tools.

However, this need for the firm to build and maintain relationships has mostly depended on the sales correspondents' behavior.

5.2.1. Sales force automation (SFA)

A Sales Force Automation (SFA) system aims to improve the quality and the speed of the information flow among the sales force, the customers and the organization in order to enhance and support relationships.

Despite the advancements in technology and the common assumption that these technologies will lead to increases in efficiency, they usually fall short, resulting in implementation failures as high as between 55 and 80%. [6]

And some of the root causes for this failure rates have been identified as the following:

- The sales organization *not accepting the SFA system* through multiple barriers such as, usage barriers, value barriers, risk barriers, tradition barriers or image barriers.
- The *lack of motivation* to do what the system enables the employees to do. Sometimes they are not even sure what the system can provide them.
- Belief that the use of *the system makes it harder* for them to continue doing their job.
- *Inertia* as the inclination to continue doing what has always worked.
- Perception of *low benefits* while the costs for deployment are high.
- *Lack of support* from the sales force organization.
- The adoption process is *time-consuming*.

Even though the previous list provides a sense of the main causes of failure while implementing an SFA system, these can be cut down to two variables:

- The **Intention**, which can be defined as the likelihood that the salesperson will adopt the technology. Intention is really important during the pre-implementation phase.
- The **Infusion** which measures the extent of the use for the system. This variable plays a very important role during the consolidation of the system within the firm.

5.2.2. SFA implementation

In order for the implementation to take place successfully, the person in charge should be either a Sales manager, a Vice president or even a Sales Force Automation manager.

However, this employee has to make sure that these new technologies are seen as a mean to increase the productivity and as a solution for some of the traditional problems faced by sales management. In other words, the sales forces must know exactly what the system is set out to accomplish.

The deployment of SFA systems are usually accompanied by major organizational changes like the shift from a cross-functional organizational process to a relationship marketing one. This implies that the sales correspondents must know the importance of sharing information related to customers, especially when it comes to retention.

Despite popular beliefs, several studies have proven that neither the size of the firm nor the size of the sales force appears to influence the potential benefits achieved by SFA systems. When successfully deployed, a Sales force automation system can provide the following benefits:

- Improved access to information by allowing information to be shared across the firm.
- Consolidation of information.
- Higher efficiency.
- Better managing and tracking of the firm's inventory.
- From the client's point of view, the communication is greatly improved.

However, SFA systems also brings a set of drawbacks:

- Lengthy training time.
- Implementation problems.
- High costs.
- Frustrated sales representatives.
- The need for constant technological updates.

5.3. Supply chain management

Supply chain management (SCM) is a very new and expanding discipline that plays a major role in enhancing productivity and profitability by smoothing the flow of resources in a supply chain.

In order to achieve its purpose, there are several strategies or technologies such as the Just in Time (JIT), Lean Production, Computer generated Enterprise resource planning and Kaizen. The aims of SCM are:

- Control the flow of a certain operating system. This may be associated with inventory control and activity scheduling across the whole range of resources, products and time limitations.
- Integrate material and information flows across the supply chain.
- Meet the board's competitive and strategic objectives by measuring organizational performance in parameters like quality, speed, dependability, flexibility and cost.
- Enhance core competitiveness. This may be achieved by developing multiple performance measures and metrics. Competitiveness ensures a defensive position over the competitors.
- Control customers as the needs and the supply chain performance may change in time.
- Help win customers and improve customer service.

From the aims we can clearly see that SCM revolves around purchasing and supply management, transportation and logistics management and production planning. However, according to *Li et al. (2006)* [7] its main activities are:

- Strategic supplier partnership. Create and maintain long-term relationships between an organization and its suppliers.
- Customer relationship. These practices are employed to manage customer complaints, build long-term relationships with them and improve their satisfaction.
- Level of information sharing. Increase both the quantity and the quality of information sharing. SCM focuses on Accuracy, timeliness, adequacy and credibility.
- Postponement. Moving forward one or more operations or activities to a much later point in the supply chain. How many and which steps to postpone must be considered.

5.3.1. SCM implementation

In order to successfully implement an SCM system, performance studies and models should be created so that the degree of achievement can be measured. While financial performance measures are important for strategic decisions, daily control of manufacturing and distribution is handled better with non-financial measures.

These metrics should be able to capture the essence of organizational performance so that the output can be measured and compared with a set of standards. These standards must be kept within a limit and remain relatively constant. Depending on the degree of influence in said measures, there are three levels:

- **Strategic level.** Measures influence from the top-level management decisions in such fields like broad base policies, corporate financial plans, competitiveness and the level of adherence to organizational goals.
- **Tactical level.** Revolves around with resource allocation and measuring performance against previously established targets in order to achieve a specific result at a strategic level. This provides feedback from mid-level management decisions.
- **Operational level.** This level requires accurate data and metrics to assess the result of decisions from low level managers.

Depending on which part of the supply chain the system evaluates, *Gunasekaran et al. (2004)* [8] introduces a list of the most common measurements and metrics used by SCM. These metrics are:

- **Metric for order planning.**
 - *Order entry method.* Determines how and to what extent are the customer specifications converted into information.
 - *Order lead time.* Measures the time between the receipt of the customer order and until the delivery of goods to the customer is finished. This is important as a measure of competitive advantage.
 - *Customer order path.* By analyzing the customer order path, activities that don't add value can be identified in order to proceed by eliminating them.
- **Evaluation of supply link.** These measures must be analyzed periodically in order to meet the firm's long-term goals. It is important at a strategic, operational and tactical level. The different measures for all of these levels are the following.
 - *Strategic level.* Lead time, quality level, cost saving initiatives, supplier pricing against market.
 - *Tactical level.* Efficiency of purchase order cycle time, booking in procedures, cash flow, quality assurance methodology and capacity flexibility.
 - *Operational level.* Ability in day to day technical representation, adherence to develop schedules, ability to avoid complaints and achievement of defect free deliveries.
- **Production level.** Performance has a direct impact on product cost, quality, speed of delivery, delivery reliability and flexibility. At production level, some of the measures can be:

- *Range of products and services.* Having more products slows the plant and reduces the likeliness to introduce new products. Manufacturers with a wide range of products are more likely to perform worse.
 - *Capacity utilization.* This affects directly the speed with which the firm responds to the customer demands.
 - *Effectiveness of scheduling techniques.* It has a direct impact on production, therefore, it affects supply chain performance.
- **Evaluation of delivery link.** This is the primary determinant of customer satisfaction. Improving delivery is always desirable to increase competitiveness. To do so, there are two measures:
 - *Delivery performance evaluation.* Increasing in delivery performance can be achieved by reducing lead time attributes. On time delivery reflect whether or not the perfect delivery has taken place, which is also a measure of customer service level. To be specific, the delivery can be evaluated by comparing the delivery date, time and conditions under which the goods were delivered, as well as by the flexibility of delivery.
 - *Total distribution cost.* As firms should try to be efficient and cost effective, they need to measure individual cost elements along with their impact in customer service. This will lead to a more effective and efficient distribution system.
- **Measuring customer service satisfaction.**
 - *Flexibility.* Establishes whether or not the firm is able to meet individual demands from customers. Flexibility may affect the development cycle, the machine or tools setup time and the production in economies of scope.
 - *Customer query time.* It corresponds to the time needed to respond a query with the required information.
 - *Post-transaction measures of customer service.* These activities play an important role in customer service and provides feedback that can be used to further improve supply chain performance.
- **Supply and chain logistics cost.**
 - *Cost associated with assets and return on investments.* The goal is to improve the productivity of the capital. Generally, firms calculate the average days required to turn cash invested into assets employed into cash collected from a customer.
 - *Information processing cost.* Corresponds to the cost associated with order entry, order updating, discounts and invoicing.

6. Sentiment analysis

Opinions are key to almost all human activities and have a huge influence in our behavior. For this reason, whenever we have to make a decision we often seek out the opinions of others. However, this is not only true for individuals, it can also be applied to organizations.

Due to the huge increase in volume of opinionated data recorded in digital form, sentiment analysis growth matches the one of social media and the World Wide Web. Sentiment analysis has spread to management sciences and social sciences due to its importance in business, finding applications in almost every organizational and social domain.

Sentiment analysis is also known as opinion mining, and it is the field of study that analyzes people's opinions, sentiments, evaluations, appraisals, attitudes and emotions towards entities such as products, services, organizations, individuals, issues, events, topics and their attributes. However, it's main focus are opinions that express or imply positive or negative sentiments.

Since the year 2000, Natural language processing (NLP) has been gaining more and more importance and it has become a very active field for research due to the wide range of applications in almost every domain. It has been widely used for data mining, Web mining and information retrieval. In fact, it has shifted from computer science to management sciences.

Businesses and organizations always want to find customer or public opinions about their own products or services, and thanks to the growth of social media, individuals and organizations are increasingly using digital data to gather public opinions. This implies that it may no longer be necessary to conduct surveys, opinion polls, and focus groups.

There are different levels of analysis depending on the granularity:

- **Document level.** At this level, the objective is to classify whether a whole opinion document expresses positive or negative sentiment. This task is commonly known as *document-level sentiment classification*. This level of analysis assumes that the document only expresses opinion on a single entity, therefore it can't be applied to documents with evaluation or comparison about multiple entities.
- **Sentence level.** The task aims to determine whether a sentence is positive, negative or neutral (Lack of opinion). This task is extremely related to *subjectivity classification*, as only objective phrases should be further analyzed.
- **Entity and aspect level.** Both at a document level and at a sentence level, the analyst does not discover what exactly the entity is. At this level, we focus on the idea that an opinion consists of a sentiment that can be either positive and negative and a target. Therefore,

identifying the sentiment without knowing the target is of limited use. Most opinions about a certain product describe different aspects, thus the aim of this level of analysis is to discover opinions and their target.

Briefly, when it comes to types of opinions, there are two main types:

- **Regular Opinions.** Express sentiment of a particular entity or a certain aspect of these entity
- **Comparative Opinions.** These opinions compare multiple entities based on shared aspects.

6.1. Concept definitions

6.1.1. Opinion definition

For a sentiment analysis solution, the definition of an opinion has to be simple, but also provide all the information needed. For a general-purpose application, it should include the following information:

- **Target.** To who/what the opinion is addressed. However, the target can often be decomposed and described with multiple levels, therefore we have to introduce two parameters, entity and aspect.
- **Entity.** The symbol “ e ” will be used to further refer to an entity.
- **Aspect.** The term used when talking about an aspect will be “ a ”.
- **Sentiment.** Sentiment can either be positive, negative or neutral, however, this can also be expressed using a numeric rating score that expresses the strength or intensity of the sentiment. Sentiment will be expressed as “ s ”.
- **Opinion holder.** Referred as “ h ”, this parameter will contain the information about who is expressing the opinion.
- **Time.** This parameter will be referred as “ t ” and corresponds to the time when the opinion was expressed.

So, an opinion will correspond to a quintuple with the following structure:

$$(e_i, a_{ij}, s_{ijkl}, h_k, t_l) \quad (\text{Eq. 6.1})$$

Where e_i is the name of the entity, a_{ij} is an aspect of e_i , s_{ijkl} is the sentiment on aspect a_{ij} of entity e_i , h_k is the opinion holder, and t_l is the time when the opinion is expressed by h_k .

Given an opinion document d , sentiment analysis consists on the process due to discover all opinion quintuples $(e_i, a_{ij}, s_{ijkl}, h_k, t_l)$ in d .

6.1.2. Entity category and Entity expression

The first component of the quintuple is the entity, to extract it, named entity recognition (NER) techniques can be applied. However, after the extraction, entities have to be categorized, as sometimes, people often write the same entity in multiple ways.

When this happens, in order to differentiate the terms entity category and entity expression were created. An *Entity Category* represents a unique entity, on the other hand, an *Entity Expression* is an actual word or phrase that indicates an entity category. So, each entity category should have its own names. The process that groups entity expressions is called *Entity Categorization*.

6.1.3. Aspect category and Aspect expression

The third component in an opinion quintuple is the aspect, and like entities, aspects can also have an *Aspect Category* that represents a certain aspect, as well as an *Aspect Expression*, which corresponds to the actual term that appears in a text and indicates an aspect category.

The process of *Aspect Categorization* is also present as the action of grouping aspect expressions. There are two types of Aspect expressions:

- **Explicit aspect expression.** Aspect expressions that are nouns and noun phrases. For example, in the sentence “*The picture quality of this camera is great*”, “*picture quality*” corresponds to the explicit aspect expression.
- **Implicit aspect expression.** Aspect expressions that are not nouns or noun phrases. The word “*expensive*” in the sentence “*This camera is expensive*” is a great example. Many of these expressions are adjectives and adverbs and are used to describe or qualify an specific aspect, nonetheless, they can also be verbs and verb phrases.

So, an entity e_i can be represented by itself as a whole or as a set of aspects $A_i = \{a_{i1}, a_{i2}, a_{i3}, \dots, a_{ij}\}$. However, this entity can be expressed with any one of its finite set of entity expressions $\{ee_{i1}, ee_{i2}, ee_{i3}, \dots, ee_{is}\}$. And each aspect can also be expressed with any one of its finite set of aspect expressions $\{ea_{ij1}, ea_{ij2}, ea_{ij3}, \dots, ea_{ijm}\}$.

A document d contains opinions on a set of entities $\{e_1, e_2, \dots, e_i\}$ and their aspects, from a set of opinion holders $\{h_1, h_2, \dots, h_k\}$, at a certain point in time.

6.1.4. Sentiment analysis tasks

Having introduced the respective terminology, given set of opinion documents D , sentiment analysis consists on the following 6 tasks:

- 1) **Entity extraction and categorization.** Extract all entity expressions in D , categorize or group them into categories that we will refer to as e_i .
- 2) **Aspect extraction and categorization.** Extract all aspect expressions of the entities (e_i) and categorize these aspects, so that each entity expression is represented by a unique aspect a_{ij} .
- 3) **Opinion holder extraction and categorization.** This consists on extracting the opinion holders (h_k), so that each group of entity and aspect has its own opinion holder.
- 4) **Time extraction and standardization.** This task consists in extracting the time when the opinion was stated and convert it to a homogeneous format.
- 5) **Aspect sentiment classification.** Determinate whether an opinion on an aspect is positive, negative or neutral. This can be done by assigning a numeric sentiment rating.
- 6) **Opinion quintuple generation.** This final task of the process produces all opinion quintuples, so it basically generates the data structure containing a single opinion from a holder about a certain aspect of an entity.

6.1.5. Types of opinions

In sentiment analysis, opinions can be classified depending on what they express (Regular or comparative opinions) or on how they express it (Explicit and implicit opinions).

- **Regular opinion.** A regular opinion is a simple and regular opinion, however it has two main types:
 - *Direct opinion.* Refers to an opinion about an entity or an aspect expressed directly. For example: *"The picture quality is great."*
 - *Indirect opinion.* Refers to an opinion expressed in an indirect way, based on its effects on other entities. This can be usually seen on the medical domain. For example: *"After taking the medicine, the bruise started hurting less"*. As we can see, the desired effect of the medicine happened on a certain bruise of the opinion holder, this statement clearly gives a positive opinion, however, the entity is the *medicine* and the aspect is the *effect on the bruise*.
- **Comparative opinion.** A comparative opinion aims to compare similarities and differences about two or more entities in order to express the opinion about a certain shared aspect. For example: *"Coca Cola tastes better than Pepsi"* or *"Coca Cola tastes the best"*. Both statements express an opinion in a *comparative* or *superlative* form.
- **Explicit opinion.** Subjective statement that gives a regular or comparative opinion.

- **Implicit opinion.** Objective statement that implies a regular or comparative opinion, however, using either a desirable or undesirable fact. For example: *"I bought a new phone last week and the battery already exploded"*. As we can clearly see, this opinion states a negative sentiment towards the phone, because the battery exploded sooner than expected.

6.1.6. Subjectivity and emotion

These two concepts are really important in sentiment analysis because they usually play important roles while expressing opinions.

An **objective sentence** presents facts about aspects, but a **subjective sentence** expresses some personal feelings, views or beliefs. An example of an objective sentence would be "iPhone is a great apple product" and an example of a subjective sentence would be "I like the new iPhone release by Apple".

Subjective sentences can come in many forms: Opinions, allegations, desires, beliefs, suspicions and speculations. The task of determining whether a sentence is subjective or objective is called *subjectivity classification*. Although it is not relevant to the scope of this project, subjective sentences may not express any sentiment and if they do, they may do it by using desirable and undesirable facts, therefore using an implicit opinion.

Emotions are our subjective feelings and thoughts and they have been classified into 6 basic emotions or categories: Love, Joy, Surprise, Anger, Sadness and Fear. Each of these basic emotions can be subdivided into many secondary and tertiary emotions, leaving us with the whole spectrum of emotions which vary in intensity.

Emotions are closely related to sentiments. They are usually used to express the strength of a sentiment. Most of the online reviews that we can find are *evaluations*, and these can be categorized as *rational evaluations* and *emotional evaluations*.

- **Rational evaluation.** From rational reasoning, tangible beliefs and utilitarian attitudes. For example: *"The sound of these speakers is clear"*.
- **Emotional evaluation.** From non-tangible and emotional responses to entities which go deep into people's state of mind. For example: *"This is the best website ever"*.

Even though it can sometimes be confusing, rational opinions are very similar to emotional opinions, however, they express no emotion.

6.2. Document sentiment classification

Document-level sentiment classification aims to consider the document as a whole basic information unit.

Problem definition: Given an opinion document d , the aim is to evaluate an entity, determine the overall sentiment s of the opinion holder about the entity. However, given the previously defined data structure for sentiment analysis, the quintuple, in this case, the entity e , the opinion holder h , and the time of opinion t are assumed known or irrelevant.

The sentiment can either take a categorical value such as positive or negative or a numeric value within a range. In the first case, the process becomes a classification problem, but on the second case, it becomes a regression problem.

In addition, to ensure that the task has certain meaning we have to assume that the opinion document d expresses opinions on single entity e and contains opinions from a single opinion holder h . This assumption can hold up for reviews of products and services because each review focuses on evaluating a single product and is written by a single reviewer.

Now we will proceed to discuss some of the most common techniques for document-level sentiment classification. The most common rely on supervised learning, however, unsupervised methods also exist. Recently some additions to this method have been introduced such as *cross-domain sentiment classification* and *cross-language sentiment classification*.

6.2.1. Supervised learning

For this application, sentiment classification essentially becomes a text classification problem, but sentiment or opinion words that indicate positive or negative opinion become far more relevant. Some examples of this words might be: *great, excellent, amazing, horrible, bad, worst, etc.*

In some instances, unigrams, which can be defined as “bag of words” with their sentiment already classified, are used as a feature for sentiment classification. These unigrams perform quite well. Some of the concepts used in supervised learning are:

- **Terms and their frequency.** Consists of individual words (unigram) and their n-grams (frequency counts for each unigram). Term frequency is traditionally used in topic-based text classification. To increase accuracy, word position may also be considered. When calculating the resulting sentiment, the TF-IDF (term frequency - inverse document frequency) weighting scheme can be used. These classifications have been proven to be highly effective for sentiment classification.

- **Part of speech (POS).** The part of speech (POS) consists on a category which a word is assigned to in accordance to its syntactic function (Noun, adjective, determiner, verb, adverb, preposition, conjunction and interjection). Adjectives tend to be important indicators of sentiment, so Part of Speech along with n-grams can be useful in order to obtain the final sentiment of the text. The following table consists of a set of POS tags:

Tag	Description	Tag	Description
CC	Coordinating conjunction	PRP\$	Possessive pronoun
CD	Cardinal number	RB	Adverb
DT	Determiner	RBR	Adverb, comparative
EX	Existential there	RBS	Adverb, superlative
FW	Foreign word	RP	Particle
IN	Preposition or subordinating conjunction	SYM	Symbol
JJ	Adjective	TO	to
JJR	Adjective, comparative	UH	Interjection
JJS	Adjective, superlative	VB	Verb, base form
LS	List item marker	VBD	Verb, past tense
MD	Modal	VBG	Verb, gerund or present participle
NN	Noun, singular or mass	VCN	Verb, past participle
NNS	Noun, plural	VBP	Verb, non-3rd person singular present
NNP	Proper noun, singular	VBZ	Verb, 3rd person singular present
NNPS	Proper noun, plural	WDT	Wh-determiner

PDT	Predeterminer	WP	Wh-pronoun
POS	Possessive ending	WP\$	Possessive wh-pronoun
PRP	Personal pronoun	WRB	Wh-adverb

Table 6.1. Part of speech tags. Source: [9]

- **Sentiment words and phrases.** Sentiment words are words used to express positive or negative sentiments. For example, *good, wonderful, amazing, bad, poor, terrible*. Most sentiment words are adjectives and adverbs; however, nouns and verbs can also be used to express sentiment. Sentiment phrases and idioms serve the same purpose as sentiment words, a clear example would be *“cost someone an arm and a leg”*, meaning that something is extremely expensive.
- **Sentiment shifters.** Expressions used to change the sentiment orientation from positive to negative or vice versa. The most common class of sentiment shifters are negation words. For example: *“I don’t like this camera”*. Bear in mind that shifters must be handled carefully as not all occurrences of such words mean sentiment change.
- **Syntactic dependency.** Word dependency can also be used to assist the measurement of sentiment of a certain document.

6.2.2. Unsupervised learning

As sentiment words are the most important factor when determining the sentiment of a certain statement, these can be used in an unsupervised manner. There’s multiple algorithms capable of doing that, however, there’s one algorithm that performs the classification based on fixed syntactic patterns (Using POS tags). This algorithm consists of three steps:

- 1) Two consecutive words are extracted if their POS tags conform to any of the patterns in the following table.

First word	Second word	Third word (Not extracted)
JJ	NN or NNS	Anything
RB, RBR or RBS	JJ	Not NN nor NNS
JJ	JJ	Not NN nor NNS

NN or NNS	JJ	Not NN nor NNS
RB, RBR, RBS	VB, VBD, VBN or VBG	Anything

Table 6.2. Pattern of POS tags to extract two-word phrases. Source: [9]

- 2) Estimates the sentiment orientation (SO) of the extracted phrases using the *pointwise mutual information* (PMI) measure:

$$PMI(term_1, term_2) = \log_2 \left(\frac{\Pr(term_1 \wedge term_2)}{\Pr(term_1)\Pr(term_2)} \right) \quad (\text{Eq. 6.2})$$

PMI allows us to measure the degree of statistical dependence between two terms. The numerator inside the Logarithm is the co-occurrence probability of term1 and term2, and the denominator is the co-occurrence probability of the two terms if they are statistically independent. Then, the SO can be obtained by associating the phrase with two reference words “Excellent” and “Poor”.

$$SO(phrase) = PMI(phrase, "excellent") - PMI(phrase, "poor") \quad (\text{Eq. 6.3})$$

The probabilities can be calculated by doing searches and recording the number of positive results. As an example, the search criteria used may be the documents that contain the words within ten words of one another in either order. The second equation can be simplified to the following:

$$SO(phrase) = \log_2 \left(\frac{hits(phrase_NEAR, "excellent")hits("poor")}{hits(phrase_NEAR, "poor")hits("excellent")} \right) \quad (\text{Eq. 6.4})$$

- 3) Given a document, the algorithm computes the average SO of all two-word extracted phrases in the document and classifies the review in accordance with to the result.

6.2.3. Sentiment rating prediction

Researchers also focused on studying the problem of predicting the rating scores. So, in this case the problem should be reformulated to a regression problem. Now we are going to introduce the most relevant methods used by researchers:

- Support vector machine (SVM) regression and multiclass classification using the one-vs-all (OVA) strategy, and a meta-learning method called metric labeling. But this method proved to be outperformed by the following.
- Modeling rating prediction as a graph-based semi-supervised learning problem using reviews with and without ratings. In the graph, each node corresponds to a document, and the link between two nodes is the similarity value between the two documents, therefore, a high similarity weight means that the two documents tend to have the same sentiment rating. The method then proceeds to improve the ratings of the unrated documents by solving an optimization problem that aims to smooth the ratings throughout the graph.
- A bag-of-opinions representation of documents to capture the strength of n-grams with opinions. In this method, opinions are defined as triples, a sentiment word, a modifier and a negator. A constrained ridge regression method learns the sentiment score or strength of each opinion from different rated reviews. Then, the algorithm does a set of statistics over the opinion scores and then uses them along with standard unigrams to predict ratings.
- For predicting the rating for each aspect, two models have to be created: Aspect model that works on individual aspects and the agreement model that focuses on the rating agreement among all aspects. Both these models were combined with learning from lexical features from already labeled documents.
- Using a Bayesian network classifier to predict ratings for each aspect in a review. In this method, instead of using a lot of reviews, they choose only those ones that evaluate aspects in a comprehensive manner to ensure that the machine learning algorithm can have enough information.

6.2.4. Cross-domain sentiment classification

Sentiment classification is highly sensitive to the domain from which the learning data is extracted. Therefore, a classifier trained using data from one domain cannot perform properly on data from another one. So, domain adaptation or transfer learning is needed in order to ease the task of switching domains. The original domain from which we extract the labeled training data will be called *source domain*, and the new domain will be called *target domain*.

Through many years and hours of research, several methods have been applied in order to tackle this problem:

- Using support vector machine (SVM) to train the classifier with either labeled data from both domains or using available data only from the target domain.
- Using Expectation-Maximization (EM) techniques for semi-supervised learning, combining small amounts of labeled data with large amounts of unlabeled data from the target domain.

- Based on feature selection proposed for transfer learning at a sentence-level. Using two fully labeled training sets from two domains, the algorithm detects common features (domain independent features) and uses them to label documents in the target domain.
- Using structural correspondence learning (SCL). Given a set of labeled reviews from the source domain and unlabeled reviews from the target domain, SCL selects common frequent features (pivot features) that will represent the shared feature space of the two domains. Then creates a correlation matrix that correlates non-pivot features with all of the pivot features. After that, singular value decomposition (SVD) is used to compute approximations. The end result is a classifier that can be used in both domains.
- Using a spectral feature alignment (SFA) algorithm to align domain-specific words from different domains into the same cluster. SFA works by constructing a graph with the domain-independent words as a set of nodes and the domain-specific words as another set of nodes. Links correspond to the frequency of co-occurrence between nodes.
- Using topic modeling to find common semantic spaces based on the domain term correspondences and co-occurrences in both domains.
- Automatically creating a sentiment sensitive thesaurus using both labeled and unlabeled documents from multiple source domains. This thesaurus is then used to expand original feature vectors to train classifiers.
- Identifying domain dependent and independent word sentiments.
- Based on the idea of label propagation on a similarity graph, each document is a node and each link is the weight computed using cosine similarity of two documents. Every original document is given a +1 (positive) or -1 (negative) and the documents on the new domain are assigned scores based on a regular sentiment classifier. The algorithm proceeds to iteratively update scores of each target domain documents by finding the nearest neighbors from the source domain and the target domain. The process stops when the scores converge.
- Using part-of-speech (POS) tags and identifying which are domain-dependent and which are domain-free.

6.2.5. Cross-language sentiment classification

Cross-Language sentiment classification aims to perform the classification of documents in multiple languages. However, researchers have always built sentiment analysis systems in their own languages, and most of it has been done in English.

However, in many cases, companies want to know and compare the customers' opinion on their products and services in different countries. To achieve this, it would require a system able to perform the task in multiple languages.

Through the years, many methodologies have been proposed and tested. Some of them have been the following. For some of the methodologies, the *Source Language* will be the one used as the basis to train the learner, and the *Target Language* will be the language in which the document we want to classify is written.

- One method starts by translating each review in the Target Language and using a lexicon-based approach to classify each translated review. This method seems to be the least precise one, however, it uses multiple translators to produce different English versions that will be further classified. The final sentiment score is calculated through an average score.
- Also, using a lexicon-based approach, a machine learner is trained with translations of labeled documents on the source language, so that it can establish connections between both languages.
- Using a co-training method and using English corpus to classify documents on the target language. The input consists of a set of labeled English reviews and a set of unlabeled reviews written in the target language. Both sets are translated into the other language so that an SVM method can be used to train a classifier.
- A method based on *structural correspondence learning* (SCL) that aims to find a set of shared features by both languages.
- Based on *multi-languages supervised latent Dirichlet allocation* (MLSLDA). This methodology classifies topics using documents from multiple languages at the same time, as it has been found that topics tend to be consistent across languages.

6.3. Sentence sentiment classification

Sentence-level sentiment classification aims to classify the sentiment expressed in each sentence. Even though there is no fundamental difference between documents and sentence level classifications, sentences usually contain a single opinion and documents can contain multiple opinions.

Problem definition: Given a sentence x , determine whether x expresses a positive, negative, or neutral opinion.

Generally, sentence-level classification is an intermediate step that can help clarify whether the opinion about a certain entity and its aspects is positive or negative. This step is usually separated into two steps:

- 1) **Classify whether the sentence contains an opinion or not.** This first step is closely related to *subjectivity classification*, a methodology that determines if a sentence expresses subjective or factual information. Even though it is more appropriate for this first step to classify each

sentence as opinionated or not opinionated, regardless of its subjectivity, objective sentences are regarded as expressing no sentiment or opinion.

- 2) **Classify the opinion in the sentence.** In order to classify sentences by sentiment, the different algorithms can make use of techniques such as: Supervised learning algorithms, Semi-supervised learning algorithms, learning algorithms like *Expectation Maximization* (EM) and *Conditional Random Fields* (CRF) and Lexicon-based methodologies.

6.3.1. Subjectivity classification

The task of subjectivity classification consists on differentiating between objective and subjective sentences.

Objective sentences are those that express some facts and information about a certain entity, while subjective sentences usually give personal views and opinions through opinions, evaluations, emotions, beliefs, speculations, judgements, allegations, stances, etc.

Generally, the methods used for subjectivity classification can be either Supervised or Unsupervised learning methods. Supervised methods tend to look for the presence of certain elements in the sentence while unsupervised methods determine the subjectivity of a sentence by comparing it to a set of already labeled data.

The term *gradability* refers to a semantic property that allows a word to be used in a comparative construct with a modifying expression that can intensify or diminish the strength of a certain attribute. Gradability is highly used in this field as gradable adjectives that express properties in different degrees of strength are good indicators of subjectivity.

A good example of subjectivity/objectivity classifiers would be an algorithm using two high precision classifiers to automatically identify subjectivity or objectivity clues. The algorithm would take a certain sentence and check whether it contains two or more subjective or objective clues. Then, depending on the number of clues, the algorithm would classify the sentence as Objective or Subjective.

An improvement for the previous algorithm would be to only use one subjectivity classifier that would look for subjectivity clues or the lack of them. Then it would classify in concordance with the results.

Subjectivity classification can classify binarily (Subjective or Objective) or through a strength scale. Two forms of subjectivity scales have been proposed so far:

- Using individual clauses down to four levels deep. The scale is: *neutral*, *low*, *medium* and *high*. Neutral indicates the absence of subjectivity and high represents the strongest degree of subjectivity.

- Using classes. The classes proposed are: S, OO, O and SN. S means subjective and evaluation, OO means positive or negative opinion implied in an objective sentence, O means objective with no opinion and SN means subjective but non-evaluative.

6.3.2. Conditional and sarcastic sentences

Conditional sentences are those that describe implications or hypothetical situations and their consequences. They usually contain two clauses: the condition clause and the consequent clause. Both these clauses are dependent on each other and their relationship can significantly impact the sentiment of the sentence.

In conditional sentences, sentiment words cannot help distinguish the opinion of a sentence, so the methods used should use some of the following linguistic features: Sentiment words/phrases and their locations, POS tags, patterns, conditional connectives, etc.

Sarcasm is a very sophisticated form of speech in which writers write the opposite of what they really mean. In sentiment analysis, when someone says something positive, he/she actually means negative, and vice versa. Sarcastic sentences are not extremely common in product reviews; however, they are very frequent in online discussions and commentaries about politics.

In order to tackle the challenge that sarcastic sentences propose, algorithms use semi-supervised learning methods that use a small set of labeled sentences that are expanded by automatically analyzing search results.

6.4. Aspect sentiment classification

Aspect-level sentiment classification aims to discover the aspects and determine whether the sentiment on each aspect is positive or negative. Even if we assume that a certain document evaluates a single entity, that does not mean that the author has a single opinion about each aspect of the entity.

At an aspect-level, the objective is to discover every quintuple $(e_i, a_{ij}, s_{ijkl}, h_k, t_l)$ in a document d . To achieve that, the following tasks have to be performed:

- 1) **Aspect extraction.**
- 2) **Aspect sentiment classification.**

Depending on the kind of application, some extra tasks may need to be performed such as grouping aspects into synonymous aspect categories. The purpose of this categorization is that each category represents a unique aspect, because people often use different words and phrases to describe the same aspect. The most common methods used to do so are:



- 1) A method based on similarity metrics defined using string similarity, synonyms and lexical distances measured using WordNet, a thesaurus dictionary that can be helpful to some extent.
- 2) A semi-supervised learning method proposed to group aspect expressions into user specified aspect-categories. This method works with labeled and unlabeled data, and uses an Expectation Maximization (EM) algorithm.
- 3) A method called Multilevel latent semantic association that groups all words into a set of topics using Latent Dirichlet Allocation (LDA).

6.4.1. Aspect extraction

In order to extract the aspects in a document d , it must be assumed that each opinion has a target, and that target is often the aspect or topic we are looking to extract from the sentence. Opinions expressed can target either explicit and implicit.

Although explicit aspect extraction has been studied extensively, limited research has been done on mapping implicit aspects to their explicit aspects. The following four approaches are the most commonly used to extract explicit aspects:

- 1) **Extraction based on frequent nouns and noun phrases.** This method aims to find nouns or noun phrases that act as explicit aspect expressions from a large number of reviews in a given domain using a data mining algorithm. Nouns and noun phrases are identified by a POS tagger, their occurrence is counted and all nouns beyond a certain frequency threshold are kept. Then, a *pointwise mutual information* (PMI) score is determined to measure the co-occurrence of a certain aspect and a discriminator.

$$PMI(a, d) = \left(\frac{hits(a \wedge d)}{hits(a) \cdot hits(d)} \right) \quad (\text{Eq. 6.5})$$

Where a is the aspect and d the discriminator.

- 2) **Extraction by exploiting opinion and target relations.** The relation between opinions and its targets can be used to extract the targets. If a sentence does not have a frequent aspect but has some sentiment words near a noun or a noun phrase, extracting that sentiment word and the nearest noun works quite well. To achieve this, a parser is used in order to identify the relation's dependency.
- 3) **Extraction using supervised learning.** If we treat this as a general information extraction problem, some supervised learning algorithms have been proposed. The most dominant methods are based on *sequential learning* using manually labeled data. Examples of the most common *sequential learning* methods are *Hidden Markov Models* (HMM) and *Conditional Random Fields* (CRF).

- 4) **Extraction using topic modeling.** This is an unsupervised learning method that assumes each document consists of a set of topics and a probability distribution over words. The output of topic modelling are word clusters that assign to each topic a probability distribution over words in the document. Two main models have been proposed: pLSA (Probabilistic Latent Semantic Analysis) and LDA (latent Dirichlet Allocation).

6.4.2. Aspect sentiment classification

In order to classify the sentiment orientation of each aspect in a sentence, the most common approaches use supervised learning or a lexicon-based method.

For the supervised learning approach, the same methods used at a sentence-level are applicable. However, the key objective is to determine the opinion on each sentiment expression, and to do so, parsing can help determine the dependency and all the relevant information.

Supervised learning methods depend on training data, and the current methods use training data from the same domain the classifier will end up targeting, as it has shown to be the most efficient approach.

The lexicon-based approach has also been shown to perform quite well in a large number of domains. This method uses sentiment lexicon, composite expressions, rules of opinions and the sentence parse tree to determine the sentence orientation on each aspect. Lexicon-based approaches have an advantage when dealing with sentiment shifters, but-clauses and all the constructs that can directly affect sentiment.

An example of a common methodology used would be the following:

- 1) **Mark sentiment words and phrases.** For each aspect contained in a sentence, this step marks all the sentiment words and phrases.
- 2) **Apply sentiment shifters.** In this step, sentiment shifters are taken into account. Sentiment shifters are words and phrases that can modify the sentiment orientation like *not*, *never*, *none*, *nobody*, *nowhere*, *neither* and *cannot*.
- 3) **Handle but-clauses.** Words or phrases that indicate opposite need special handling because they can change the sentiment orientation too. The sentiment orientation before the contrary word and after the contrary word are opposite to each other so, if the opinion cannot be determined in one side, we could assume that on the other side it would just be the contrary.
- 4) **Aggregate opinions.** By applying an opinion aggregation function to the sentiment scores, we can determine the final orientation of the sentiment on each aspect in the sentence. The sentiment orientation for each aspect a_i in the sentence s is determined by the following function

$$score(a_i, s) = \sum_{sw_j \in s} \frac{sw_j.so}{dist(sw_j, a_i)} \quad (\text{Eq. 6.6})$$

Where sw_j is a sentiment word/phrase in s , $dist(sw_j, a_i)$ is the distance between the aspect a_i and the sentiment word sw_j , and $sw_j.so$ is the sentiment score of sw_j .

6.5. Opinion summarization

In every sentiment analysis application, the analyst needs to study opinions from many people because looking at only the opinion of a single person is usually insufficient; therefore, a summary is needed.

Aspect-based summarization is the most commonly used technique, as it is widely applied in industry in systems such Microsoft Bing and Google search products. Target-based opinion summarization captures the essence of opinions (targets and sentiments) and also give a quantitative view of how many people have positive and negative opinions about each aspect. This can be easily represented in a bar chart.

If time is also extracted, one can show the trend of opinions on different aspects, even if sentiment is not found, the frequency value of each aspect can be useful to find out which aspects users are more concerned about.

Contrastive view summarization aims to find opposite viewpoints and compare them. Given a positive and a negative set of sentences, this summary starts by extracting a number of contrastive pairs of sentences (both sentences must have opposite opinions about the same aspect). To achieve this, mining techniques are used to extract the sentences and pair them respectively.

Traditional summarization is also available, but as it produces short text summaries without taking into account aspects and sentiments and there is not a quantitative perspective, it is assumed to be a really weak summarization method.

6.6. Opinion spam detection

Opinions from social media sites are increasingly used by individuals and organizations for making purchase decisions and making choices or even marketing product design. Fake opinions to promote or discredit some products, services, organizations, individuals, and even ideas without disclosing their

real intentions. These individuals are called *opinion spammers* and their activities are called *opinion spamming*.

As opinions in social media are becoming more and more useful, opinion spamming will become more rampant and sophisticated, which will present a major challenge to detect it. Nonetheless, these opinions must be detected in order to ensure that using social media as a source of sentiment information is liable and trustworthy.

The true challenge of opinion spam detection is that unlike any other form of spam, it is very hard to detect, if not impossible to recognize. However, some algorithms use certain data in order to recognize certain common patterns between opinion spammers. For example, fake reviewers like to use *I, myself, mine*, etc. to give readers the impression that their reviews express their true experiences.

There are three types of spam reviews:

- **Fake reviews.** These are untruthful reviews that are written not based on the genuine reviewer's experience, but because of hidden motives. These motives could easily be to promote a certain entity or just to damage its reputation.
- **Reviews about brands only.** These are reviews that do not comment on the specific product or service that they are supposed to review. Instead, they are the reviewer's opinion on the brand or manufacturer of the product itself.
- **Non-reviews.** These are not reviews that contain either advertisements or irrelevant text with no opinion.

The last two types of spam reviews are easy to detect; however, the research is currently being focused on the first type, because of its difficulty when trying to detect it.

The following table includes a set of labels that we will be using to refer to certain types of reviews, depending on the product.

	Positive fake reviews	Negative Fake reviews
Good quality product	A	B
Average quality product	C	D
Bad quality product	E	F

Table 6.3. Fake reviews labelled depending on the product. Source: [9]

Not all fake reviews are equally harmful. By assuming that we know the true quality of the product, fake reviews labeled as A and F are not really harmful, however, the ones labeled as B, C, D and E can have a major harmful impact.

6.6.1. Individual vs group spamming

Fake reviews can be written by many types of people, from regular consumers to businesses that provide fake review writing services. Generally, spammers can either work individually or as a member of a group.

- **Individual spammers.** Writing fake reviews alone by him/herself using a single user-id.
- **Group spammers.** There are two sub-classes:
 - A group of different people trying to promote a target or damage its reputation.
 - A single person registered with multiple user-ids. Even though it is just one person writing the reviews, the multiple user-ids behave just like a group.

Group spamming potentially harmful because the group can take control of the sentiment on a product and completely mislead potential customers.

6.6.2. Types of Data

There are three main types of data that we can extract from a review, in order to be used:

- **Review content.** From the actual text of the review we can extract data such as linguistic features, POS n-grams and other syntactic and semantic clues that can help us conclude.
- **Meta-data about the review.** Like the star rating, the user-id of the review, the time when it was posted, the time taken to write it, the IP address of the computer, geolocation, etc. From these kind of data, we can find behavioral patterns of reviewers and their reviews.
- **Product information.** Basically, the product information such as price, description, sales volume, etc.

6.6.3. Supervised spam detection

The opinion spam detection can be formulated as a classification problem with two possible classes, *fake* and *non-fake*. However, there is no reliable fake review and non-fake review data available to train a machine learning algorithm, but several detection algorithms have been proposed.

Since writing elaborate reviews is not an easy task, spammers use the same reviews or slightly modified reviews for different products. These duplicates can be divided into four categories:

- Duplicates from the same user-id on the same product.
- Duplicates from different user-ids on the same product.
- Duplicates from the same user-id on different products.
- Duplicates from different user-ids on different products.

Three types of features are used in training the machine learning algorithms:

- **Review centric features.** These include features about each review like words, n-grams, the number of times that brands are mentioned, percent of opinion words, the length of the review and the number of helpful feedback.
- **Reviewer centric features.** These include features about each reviewer like average rating given by the reviewer, the mean and standard deviation on the rating, the ratio of the number of reviews that this reviewer wrote which were first reviews of products, the total number of reviews that the reviewer has written and the number of cases in which he/she was the only reviewer.
- **Product centric features.** These include features about the product like the price, the sales rank, the mean and the standard deviation of review ratings of the product.

The results showed the following results:

- Ratings with significant negative deviations from the average rating of a product tend to be fake. However, positive deviations do not follow that same tendency as much as the negative ones.
- Reviews that are the only reviews of some product are likely to be fake.
- Top-ranked reviewers are more likely to be fake reviewers. Some top reviewers wrote thousands or even tens of thousands of reviews, which is unlikely for an ordinary consumer.
- Fake reviews tend to get good feedback and genuine reviews leap towards getting bad feedback.
- Products of lower sales ranks are more likely to be spammed.

6.6.4. Unsupervised spam detection

Due to the difficulty of manually labeling of training data, unsupervised spam detection alone is difficult to implement. Nonetheless two main approaches have been developed.

- **Spam detection based on Atypical Behaviors.** In order to identify spammers by looking at abnormal behaviors based on different review patterns. Each model assigns a behavior score to a reviewer by measuring the extent to which its behavior correlates with the one of a spammer. The spamming behavior models are:

- **Targeting products.** A spammer will direct most of his efforts on promoting or victimizing the target products. The spammer is expected to monitor the products closely and mitigate the ratings by writing fake reviews when appropriate.
- **Targeting groups.** This model defines the pattern of spammers aiming to manipulate a certain rating within a short time span.
- **Generating rating deviation.** As a genuine reviewer is expected to give ratings similar to other raters of the same product. If spammers try to promote or demote some products, their ratings deviate a lot from those of other reviewers.
- **Early rating deviation.** Early deviations in ratings are a very common behavior of spammers. These reviews are more likely to attract attention from other reviewers.

As an example, the behaviors that the model looks for are high ratings for products of a brand while the other reviews are generally negative, reviewers with multiple reviews for a single product, whether or not the most positive reviews for a brand of products are written by only one reviewer or reviewers who write only positive reviews to one brand and only negative reviews to another brand.

- **Spam detection using Review Graph.** This is a graph model proposed to detect spam in online store reviews. There are three types of nodes, reviewers, reviews and products. A reviewer node has a link to each review that he/she has written. A review node is linked to a product if the review is about that product. Then, each node is attached to a set of features such as *trustiness* of reviewers, *honesty* of review and *reliability* of the product. A reviewer is more trustworthy if he/she has written more honest reviews; a product is more reliable if it has more positive reviews from trustworthy reviewers and a review is more honest if it is supported by many other honest reviews. These relations define a graph that can be computed in order to find the values assigned to each node. These values allow us to rank them.

6.6.5. Group spam detection

A group spam detection algorithm was proposed and it works in two steps:

- 1) **Frequent pattern mining.** Process the data in order to produce a set of transactions. Each transaction represents a unique product that consists of all the user-ids who has reviewed a certain product. Then, a mining algorithm groups reviewers who have reviewed the same set of products. These groups are regarded as candidates for spam groups.
- 2) **Rank groups based on a set of indicators.** This uses a set of indicators to catch unusual group and individual behaviors like writing reviews in short time windows, right after the product launch, content similarity, rating deviation, etc.

7. Twitter sentiment analysis

In order to showcase the strong capabilities and simplicity of sentiment analysis as a Business Intelligence tool aiming to ease the decision-making process. I've built a set of applications that can be executed in order to perform various tasks with a set of twitter data.

However, during the development of the project my view has shifted many times because multiple reasons such as, technological bottlenecks, complexity and lack of programming knowledge.

My initial view was that of a system able to perform the following tasks:

- Capture Twitter data in real time, given a set of filters.
- Process and analyze the data and keep updating the results.
- Showing the results in a set of Plots/Graphs/Maps that allow the decision maker to extract certain conclusions on the opinion on social media about a certain topic.

The first obstacle encountered was the technical difficulties to implement a system that would be capturing the data, analyzing it and creating a visual representation of the results in real-time. So the decision was made to opt for a modular solution. The main functionalities would be separated into two categories so that they could be launched separately in order to serve the same purpose as it was originally intended. These two module categories are:

- **Data capture.** This module captures the twitter data and saves it into a file so that it can be analyzed afterwards.
- **Data Analysis and visualization.** This category is integrated by multiple modules that analyze the file with the data. Then it shows the result through a graph, bar plot or map.

7.1. Tasks and planification

In order to give an insight of all the tasks that have been done throughout the development of this project, I have to start describing the starting point. Obviously, the scope of the project requires a set of programming skills and abilities to make it easier to follow and learn about the task of analyzing datasets.

This required me to acquire a certain degree of familiarity with the chosen language and the field's terminology. Given that my programming skills were mostly basic, the tasks chosen in order to develop the process were the following: Field Research, learning to program and Development of the solution.

7.1.1. Field research

This was the most extensive and tedious task of the project. It basically consisted of reading paper after paper on multiple fields where the project could fit in. The fields chosen to be related with the projects are:

- **Decision Making.** As the whole purpose of the project is to develop a tool that can be used to assist in the process of decision making, it was set as a requirement to study the theory behind the making of a decision, alongside with multiple models that simplify the task.
- **Business Intelligence.** This field of study was found to be the extremely related to the project itself. As the whole purpose of the field is to create a set of tools that not only ease the task of managing enterprises and businesses, but also automate many tasks by analyzing either internal or external datasets of structured data
- **Data Mining.** Any application that requires of a dataset in order to further analyze it, also needs to use certain data mining features. Given that this project aimed at analyzing as much tweets as possible in order to extract certain information, Data Mining was set to be highly related and potentially useful.
- **Big Data.** Big data is a relatively new field that aims to analyze enormous amounts of data that is generated in a daily basis. Even though the goal of this project is to deal with datasets much smaller than the ones used in Big Data applications. All the procedures and terminologies were found to be useful because the only difference found was the scale.
- **Sentiment analysis.** As the application developed aimed to extract the opinion of a text shorter than 140 characters. All the models and procedures used in sentiment analysis have been extremely useful in order to understand how sentiment classification algorithms work.

7.1.2. Learning to program

Even though my initial programming skills were enough to write simple lines of code and loops, the application required me to learn how to use and interact with multiple libraries.

The purpose of these libraries is to ease the process of actions such as reading/writing data on files, connecting with the twitter servers in order to generate an incoming stream of data or even analyzing the twitter data using a natural language toolkit (NLTK).

Besides learning how to use libraries, I also had to schedule multiple tests regarding functionalities and data structure.

The most commonly used and important libraries for the project are:

- Natural Language Toolkit (NLTK).

- Tweepy.
- Matplotlib.
- Gmaps.
- Jupyter notebook.

7.1.3. Development of the solution

This task consisted of constructing the code that would end up covering all the module's functionalities. For the "Data capture" functionality, the procedure followed consisted of choosing a data capturing method, choosing a data structure that would be able to serve the purpose and select the language or languages it would be able to work with.

I also had to decide which modules would be necessary and what purpose would they serve when analyzing the data. Therefore, the design process for each module consisted of:

- Establish the purpose of the module.
- Design a functional structure.
- Perform tests with a small set of data.
- Solve possible problems preventing the system to work properly.

7.2. Tools

To achieve the purpose of the project, a certain set of tools had to be used, for instance, the language chosen to program all the modules was Python, and some of the most useful libraries were NLTK and Tweepy.

7.2.1. Python

The main reason why I chose python as the programming language for the project was because of its versatility due to being one of the most common and used languages nowadays. However, some of the reasons why it is one of the most widespread languages are:

- Python is a very robust language, solid and powerful. It has a relatively small quantity of lines of code, which makes it easier to debug, perform maintenance and it's less prone to issues.
- Python is flexible due to the number of fields it is compatible with such as web applications development, desktop apps. Furthermore, it has become a staple in the scientific community.
- Python is easy to learn and use due to its simplicity and the enormous amount of online documentation and guides to follow.

- Python is an open source dynamic language that's excellent for enterprise applications. Also, being one of the languages with more libraries gives developers a set of tools that no other language can provide.

7.2.2. Natural Language Toolkit

Natural language toolkit (NLTK) is the leading platform to build python programs that work with human language data. It provides a fairly easy to use interface and a lot of corpora and lexical resources such as WordNet and a set of text processing libraries for classification, tokenization, stemming, tagging, parsing, semantic reasoning, etc.

For the project, we decided to only use one of NLTK's modules, the "nltk.sentiment.vader", which is a parsimonious rule-based model for sentiment analysis of social media text.

VADER stands for *Valence Aware Dictionary for sEntiment Reasoning*, but it distinguishes itself from other industry standards because it is more sensitive to sentiment expressions in social media contexts. It provides the user with a sentiment score between -1 and 1. Being 0 a neutral score, -1 a completely negative statement and 1 an overwhelmingly positive statement.

In [10], VADER was found to be the most effective tool to extract sentiment from tweets with an accuracy of 96%. The following table shows the results obtained over the study.

3-Class classification Accuracy (F1 scores)				
	Tweet	Movie	Amazon	NYT
VADER	0.96	0.61	0.63	0.55
NB (tweets)	0.84	0.53	0.53	0.42
ME (tweets)	0.83	0.56	0.58	0.45
SVM-C (tweets)	0.83	0.56	0.55	0.46
SVM-R (tweets)	0.65	0.49	0.51	0.46
NB (movie)	0.56	0.75	0.49	0.44
ME (movie)	0.56	0.75	0.51	0.45

NB (amazon)	0.69	0.55	0.61	0.48
ME (amazon)	0.67	0.55	0.60	0.43
SVM-C (amazon)	0.64	0.55	0.58	0.42
SVM-R (amazon)	0.64	0.49	0.48	0.44
ONB (nyt)	0.59	0.56	0.51	0.49
ME(nyt)	0.58	0.55	0.51	0.50

Table 7.1. Three-class accuracy for each machine trained model as testes against every other domain. Source: [10]

7.2.3. Tweepy

This library provides easy access to all the whole twitter API. One of the main usage cases for tweepy is monitoring tweets and doing actions when something happens. This allows us to create an object called *StreamListener*, which will monitor tweets in real time and catch them depending on some filters and how it is set to work.

Talking everything into account, tweepy is a great open-source library that provides access to the Twitter API for Python. Thus, it is considered to be the best Twitter library for python.

7.2.4. Other tools used

Some of the other relevant tools used were the Matplotlib library, Gmaps and Jupyter:

- **Matplotlib.** This is a python 2D plotting library which aims to ease the process of data plotting and representation. For simple plotting, there's a module called *pyplot* that provides a MATLAB-like interface. The main reason why Matplotlib was chosen as the main representation tool is because all the details of the plot can be edited, and it works well with all the most important numerical tools in python.
- **Gmaps.** This plugin allows the user to interact with the Google Maps API. This plugin was chosen because it allowed a great variety of data representation over the map such as: Heat Maps, Choropleth maps and simple markers and symbols. The only downside to this plugin is that it requires of another web client called *Jupyter notebook*.
- **Jupyter Notebook.** This is a server client that allows editing and running notebooks documents via a web browser. It can be executed in a local desktop requiring no internet access.

7.3. Code

As it has already stated before, the structure of the functionalities and modules of the application build are.

- Data Capture.
- Data Analysis and visualization.
 - Added Accumulative sentiment.
 - Accumulative average sentiment.
 - Sentiment degree frequency.
 - Timely frequency.
 - Number of positive and negative tweets.
 - Number of tweets by country.
 - Number of positive/negative tweets by country.
 - Geolocation of positive/negative tweets.

The Code for each of the application modules will be found in the **annex**.

7.3.1. Data capture

This functionality allows us to access the twitter API through the *tweepy* library and save all the data on a .json file through the *json* library.

When the code found in the **annex A** is executed, the file created or updated will contain a set of tweets in the following Json data structure:

- Created_at
- Id
- Id_dtr
- Text
- Source
- Truncated
- In_reply_to_status_id
- in_reply_to_status_id_str
- In_reply_to_user_id
- In_reply_to_user_id_str
- In_reply_to_screen_name
- User: { }
- Geo

- Coordinates
- Place: { }
- Contributors
- Retweeted_status

7.3.2. Added accumulative sentiment

For each tweet that is not a retweet, this module calculates the sentiment and adds the resulting value to a variable. Then, Matplotlib is used to plot the variation of the variable for each tweet analyzed. The code can be found in the **annex A**.

For a sample of data containing over 3000 tweets captured through various instances we get the following result:

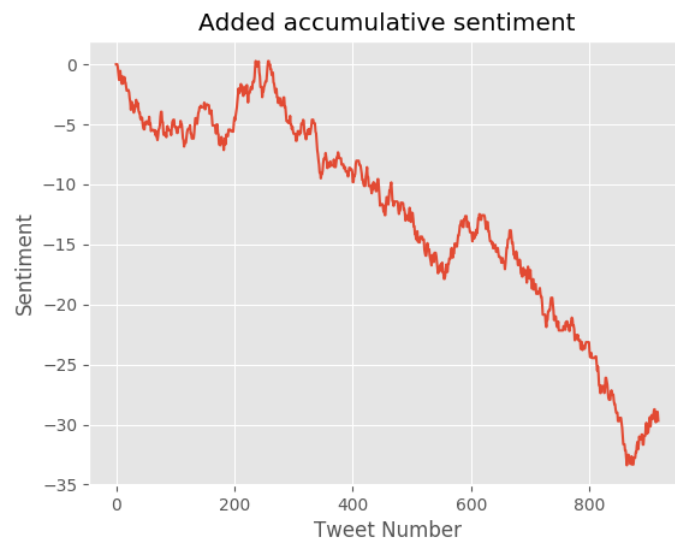


Figure 7.1. Sample result of the “added accumulative sentiment” algorithm.

7.3.3. Accumulative average sentiment

For each tweet that is not a retweet, the module computes the accumulated average sentiment score. Then Matplotlib is used to plot how the variable changes for each tweets analyzed. The code can be found in the **annex A**.

For a sample of data containing over 3000 tweets captured through various instances we get the following result:

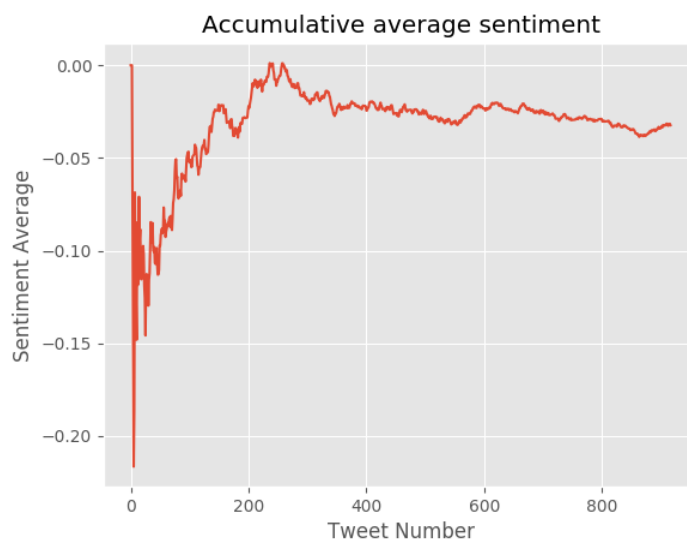


Figure 7.2. Sample result of the “accumulative average sentiment” algorithm.

7.3.4. Sentiment degree frequency

For each tweet that is not a retweet, this module analyzes the text, rounds the score and saves it into a table. Then, a counter is used to count how many times each value appears on the table. Then, Matplotlib is used to show that information using a Bar plot. The code can be found in the **annex A**.

For a sample of data containing over 3000 tweets captured through various instances we get the following result:

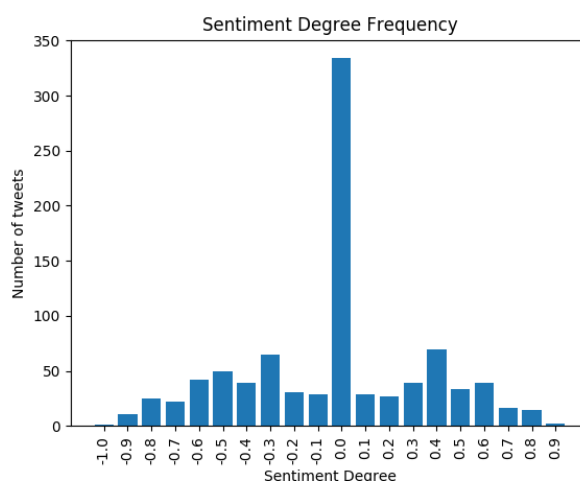


Figure 7.3. Sample result of the “sentiment degree frequency” algorithm.

7.3.5. Timely frequency

For each tweet that is not a retweet, this module records the time into a table (Without taking into account seconds). Then, a counter is used to count how many instances of each time are there, and matplotlib is used to plot the information. The code can be found in the **annex A**.

For a sample of data containing over 3000 tweets captured through various instances we get the following result:

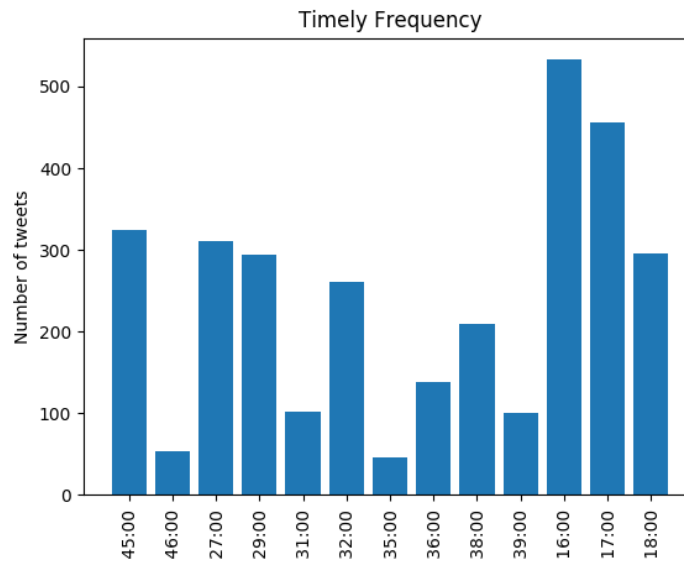


Figure 7.4. Sample result of the “timely frequency” algorithm.

7.3.6. Number of positive and negative tweets

For each tweets that is not a retweet, this module calculates the sentiment score and counts how many positive and negative results we get. Neural values are not taken into account. Then, matplotlib is used to plot the information. The code can be found in the **annex A**.

For a sample of data containing over 3000 tweets captured through various instances we get the following result:

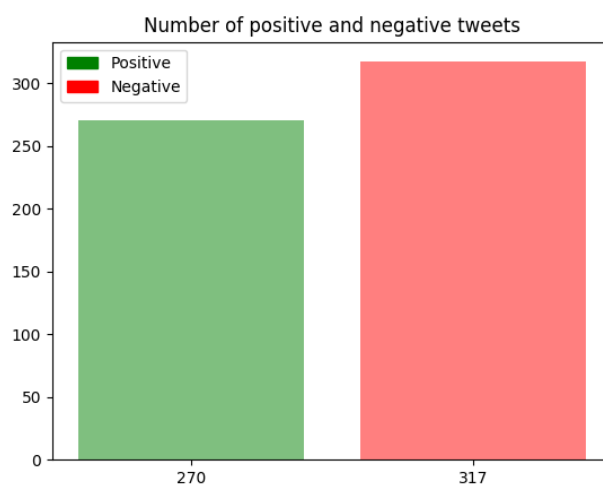


Figure 7.5. Sample result of the “timely frequency” algorithm.

7.3.7. Number of tweets by country

For each tweets that is not a retweet, and has geolocation enabled, this module extracts the country from which the tweet was made, and then counts how many tweets are made from each one of the instances. Matplotlib is used to plot using a bar plot. The code can be found in the **annex A**.

For a sample of data containing over 3000 tweets captured through various instances we get the following result:

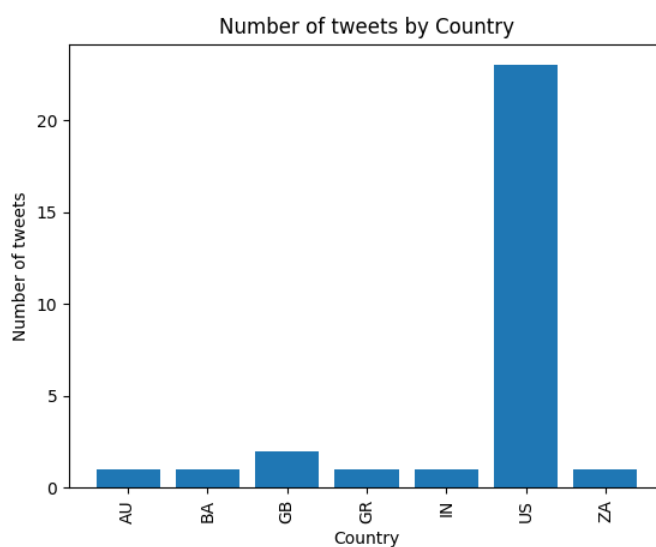


Figure 7.6. Sample result of the “number of tweets by country” algorithm.

7.3.8. Number of positive/negative tweets by country

For each tweet that is not a retweet and has geolocation enabled, this module calculates the sentiment score and can either plot the amount of positive or negative tweets using matplotlib. The code can be found in the **annex A**.

For a sample of data containing over 3000 tweets captured through various instances we get the following result:

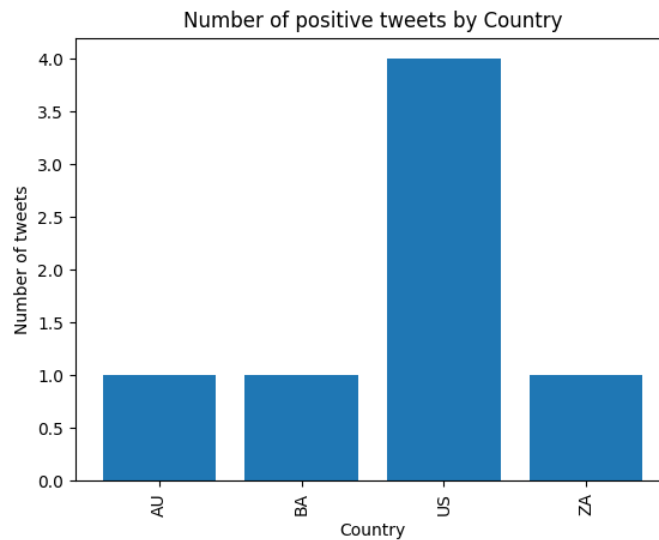


Figure 7.7. Sample result of the “Number of positive/negative tweets by country” algorithm.

7.3.9. Geolocation of positive/negative tweets

For each tweet that is not a retweet, that has some geographical coordinates linked to it, this module computes the sentiment score and plots on top of a map, the locations from where the tweets have been done. Gmaps and Jupyter notebooks is used in order to execute the code and plot using the Google maps API. The code can be found in the **annex A**.

For a sample of data containing over 3000 tweets captured through various instances we get the following result:

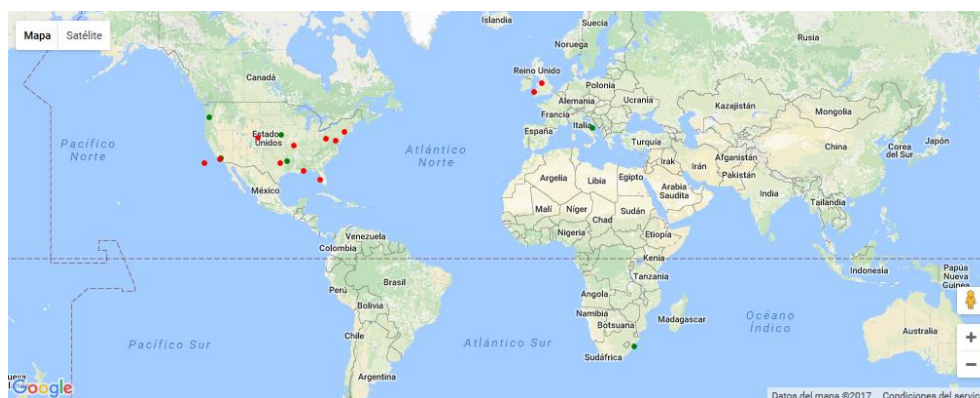


Figure 7.8. Sample result of the “geolocation of positive/negative tweets” algorithm.

8. Results

In order to perform a complete analysis using the code previously described, a data capture was performed on Saturday May 6th from 14:55 to 15:22. As the objective here was to obtain the maximum number of tweets in the shortest amount of time, the term used to filter the tweets was “Trump”.

The goal of this analysis is not to get into any political stance, but to provide some results given a fairly large dataset of tweets, and the current US president was considered to be a topic controversial enough to satisfy

The resulting *data.json* file ended up containing 32.574 tweets and retweets, weighting 201,575 MB.

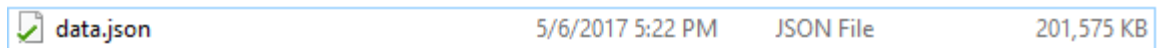


Figure 8.1. Properties of the JSON file containing all the data captured.

```
{
  "created_at": "Sat May 06 15:22:24 +0000 2017",
  "id": "860877415614205952",
  "id_str": "860877415614205952",
  "text": "RT @matthewam: ",
  "created_at": "Sat May 06 15:22:25 +0000 2017",
  "id": "860877416960598017",
  "id_str": "860877416960598017",
  "text": "@brianstelter ",
  "created_at": "Sat May 06 15:22:25 +0000 2017",
  "id": "860877416956391424",
  "id_str": "860877416956391424",
  "text": "RT @kwillil104: ",
  "created_at": "Sat May 06 15:22:25 +0000 2017",
  "id": "860877417879146500",
  "id_str": "860877417879146500",
  "text": "RT @MariaMHRW: ",
  "created_at": "Sat May 06 15:22:25 +0000 2017",
  "id": "860877417283375104",
  "id_str": "860877417283375104",
  "text": "THE TRUMP WHI: ",
  "created_at": "Sat May 06 15:22:25 +0000 2017",
  "id": "860877416893472770",
  "id_str": "860877416893472770",
  "text": "RT @tonyposna: ",
  "created_at": "Sat May 06 15:22:25 +0000 2017",
  "id": "860877417530970113",
  "id_str": "860877417530970113",
  "text": "RT @maggieNYT: ",
  "created_at": "Sat May 06 15:22:25 +0000 2017",
  "id": "860877416667000832",
  "id_str": "860877416667000832",
  "text": "RT @20committ: "
}
```

Figure 8.2. View of the JSON file containing all the data captured.

8.1. Added accumulative sentiment

The result after computing the *added accumulative sentiment* algorithm on the *data.json* file is:

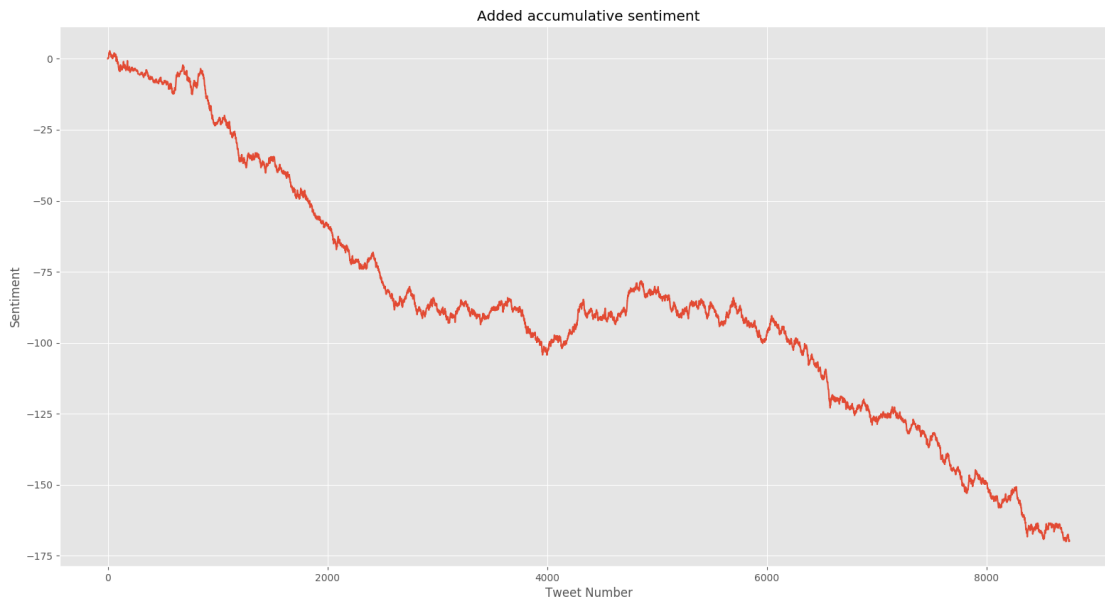


Figure 8.3. Resulting graph of the added accumulative sentiment algorithm.

As the algorithm keeps adding the sentiment values of all the tweets as it analyzes all of them, this graph provides us with an absolute value of variation. In this case, a negative tendency can clearly be seen, which brings us to conclude that the general public using twitter think negatively about the chosen topic.

This graph not only provides information about the sentiment degree, but also how it evolves from tweet to tweet.

8.2. Accumulative average sentiment

The result after computing the *accumulative average sentiment* algorithm on the *data.json* file is:

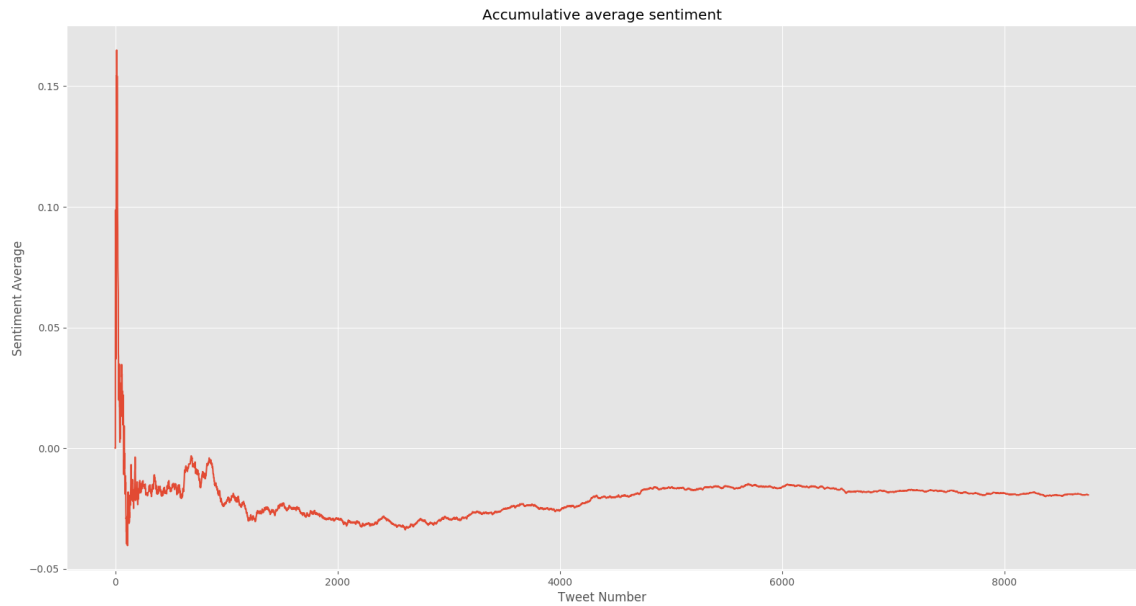


Figure 8.4. Resulting graph of the accumulative average sentiment algorithm.

The accumulative average sentiment algorithm aims to find an average sentiment value for a given tweet and all the previous ones. Given the results obtained, the negative tendency can also be seen. However, in this case, we can also see that after a little over 8500 tweets the average value is around -0.02. This means that even though there's a general negative sentiment, it is just slightly negative.

8.3. Sentiment degree frequency

The result after computing the *sentiment degree frequency* algorithm on the *data.json* file is:

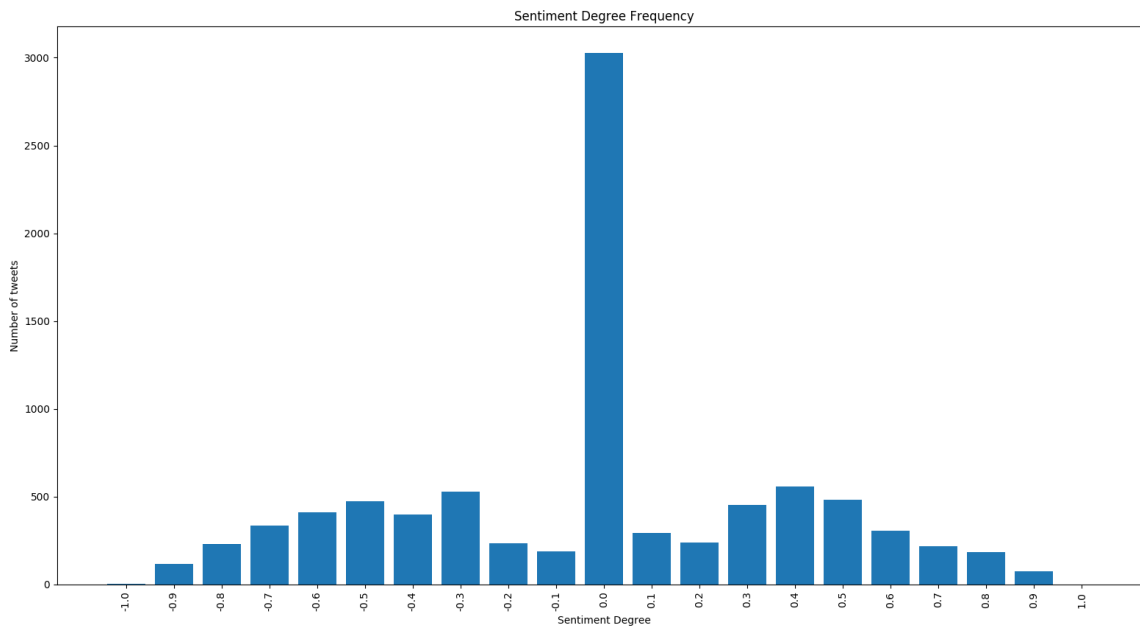


Figure 8.5. Resulting bar plot of the sentiment degree frequency algorithm.

In this case, the algorithm uses a bar plot to show how many tweets have rounded sentiment values for each value between -1.0 and 1.0. From the results obtained we can conclude that a lot of twitter users talking about the chosen topic don't express any sentiment at all. However, for the rest of degrees are distributed as it would expected, showing peaks of -0.3 and 0.4 for both negative and positive tweets respectively.

8.4. Timely frequency

The result after computing the *timely frequency* algorithm on the *data.json* file is:

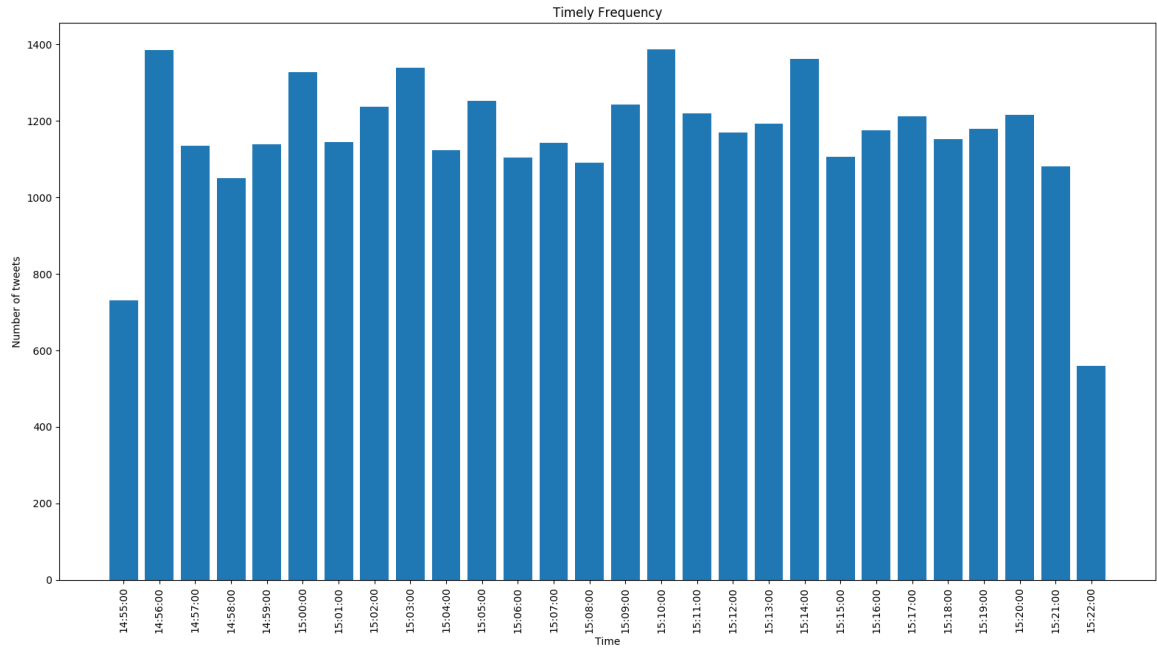


Figure 8.6. Resulting bar plot of the timely frequency algorithm.

In this case, the plot bar shows the number of tweets posted for each minute the data capture process was running. In this case, there's not much useful information to be extracted. Therefore, in order to use this algorithm to its full potential, the data capture should be running for 24 hours and the bar plot should show the number of tweets each hour.

8.5. Number of positive and negative tweets

The result after computing *the number of positive and negative tweets* algorithm on the *data.json* file is:

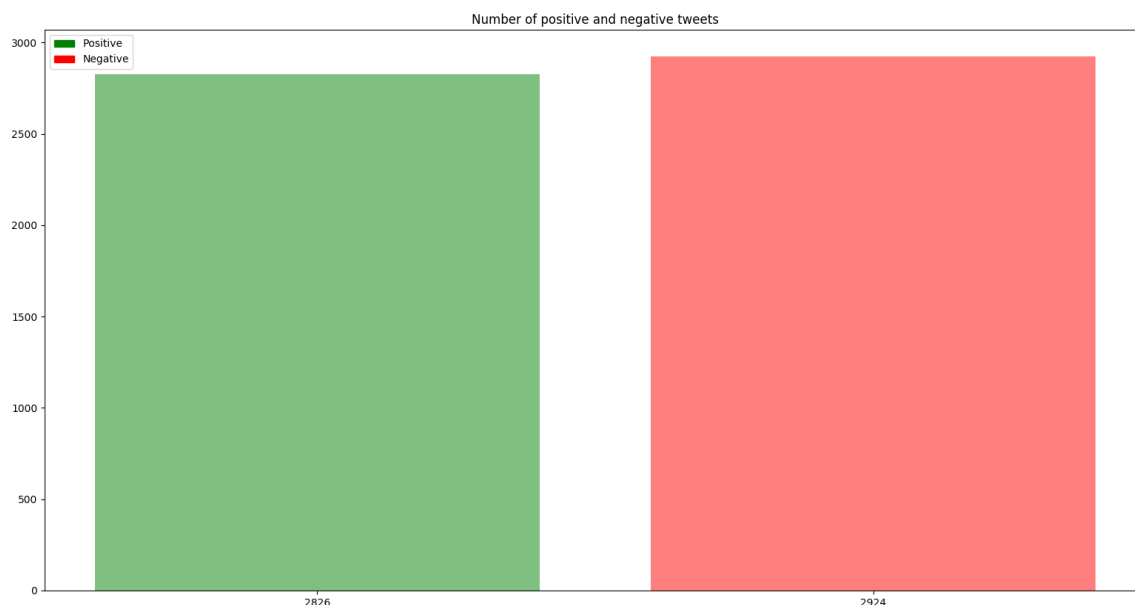


Figure 8.7. Resulting bar plot of the number of positive and negative tweets algorithm.

In the previous figure, we can see that there is a slight difference between the number of positive and negative tweets. Nonetheless, this slight difference can justify the negative tendency found on both **Figure 8.3** and **Figure 8.4**.

8.6. Number of tweets by country

The result after computing the *number of tweets by country* algorithm on the *data.json* file is:

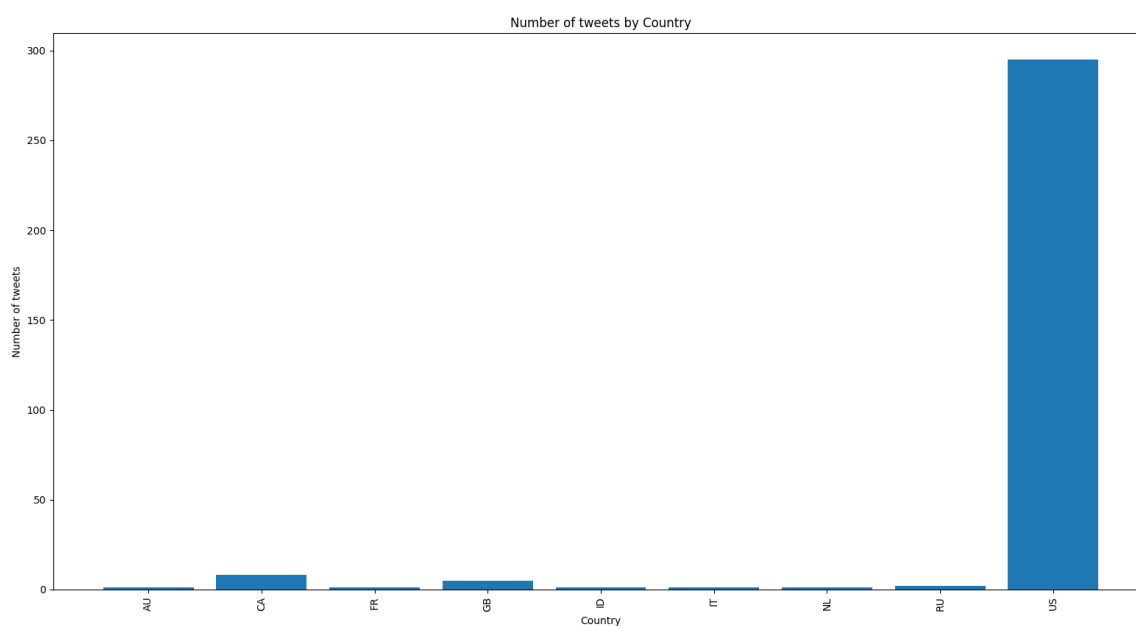


Figure 8.8. Resulting bar plot of the number of tweets by country algorithm.

As expected, the United states (US) is the country with the highest number of tweets. But from the rest of the countries, Canada (CA), Great Britain (GB) and Russia (RU) are following behind.

8.7. Number of positive/negative tweets by country

The result after computing the *number of positive/negative tweets by country* algorithm on the *data.json* file is:

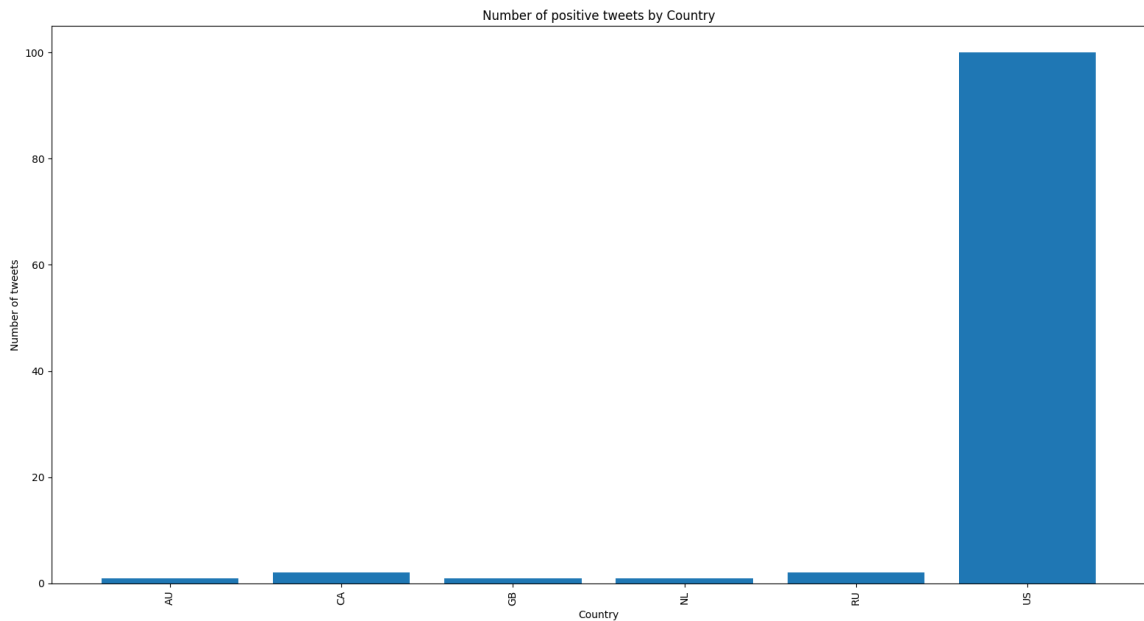


Figure 8.9. Resulting bar plot of the number of positive tweets by country algorithm.

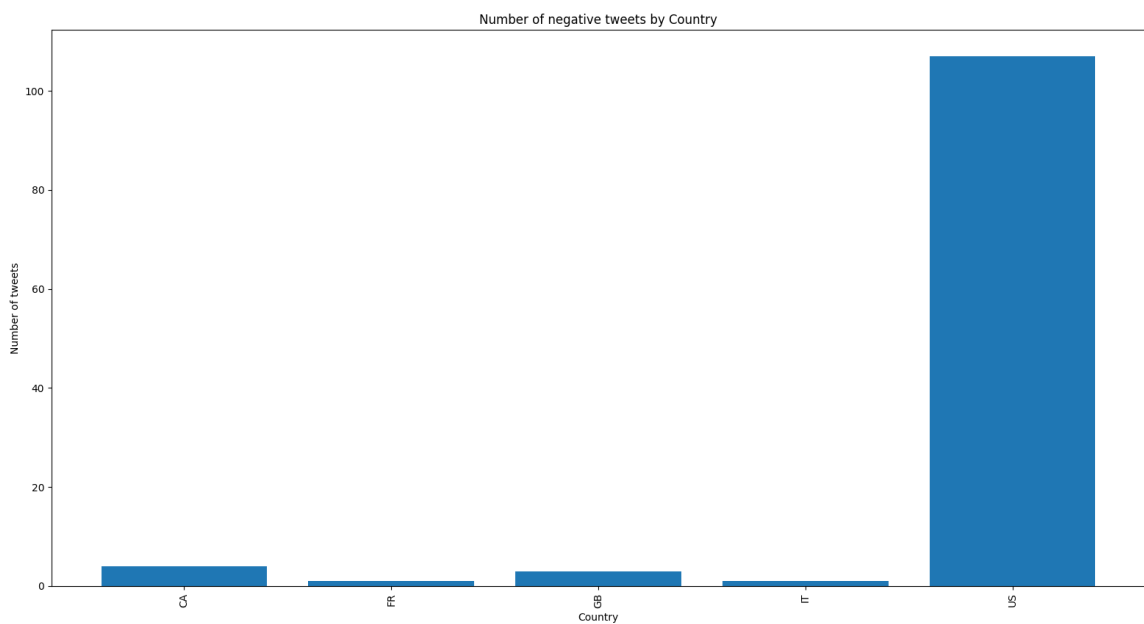


Figure 8.10. Resulting bar plot of the number of negative tweets by country algorithm.

From all the tweets shown in **Figure 8.8**, the two previous figures separate them into positive and negative showing some interesting results. For the US, the number of positive/negative tweets is evenly distributed. However, countries like GB and CA have much more negative tweets than positive ones. Surprisingly, RU has only shown positive sentiment toward the chosen topic.

8.8. Geolocation of positive/negative tweets

The result after computing the *geolocation of positive/negative tweets* algorithm on the *data.json* file is:

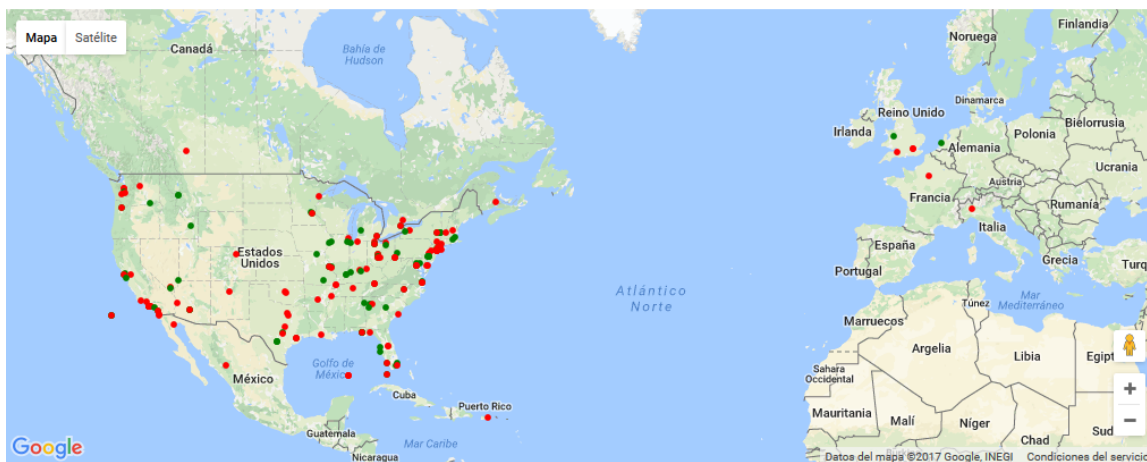


Figure 8.11. Resulting Map view of the Geolocation of positive/negative tweets algorithm.

This algorithm is only able to process tweets with geo-location enabled. From the small number of tweets that satisfy the condition, there's a clear negative dominance in the US. Due to the lack of tweets coming from the EU, no conclusions can be extracted from them.

Conclusions

To fully extract a conclusion about whatever this project has led to, two aspects will have to be considered, the original goal of the project and the usefulness of Big Data as a tool to help decision makers.

Fulfillment of the original goal

As previously explained, the, main objective of the project was to build a sentiment analysis application and to learn more about a rapidly growing field known as Big Data.

As to the sentiment analysis application, the goal has not been fully achieved because not only the original goal was ambitious and complicated, but some technical difficulties have made it even harder. Nonetheless, the final result is capable of providing analysis of a set of previously captured data along with a visual representation. The only drastic differences are introduction of modularity in order to simplify the process, and the lack of live updating graphs and charts.

Concerning Big Data as a field of study, by being involved in developing an application and reading a lot of articles about it, I've been able to understand more about it, as well as figure out why has it become one of the fastest growing markets in the last few years.

Big Data as a tool for decision makers

Regarding whether or not Big Data can be used as a tool to assist a decision-making process, we have to acknowledge that Big Data is far bigger than we can imagine. It has presence in almost every field as a tool to perform data analysis of really big datasets.

By analyzing the data, it provides us with conclusions such as behavioral patterns, the identification of new market segments or even a summarization of customer feedback. All this information can be used by a decision maker to his/her advantage in order to mold and adapt the business model to the customers' needs.

The application build for this project proves that there is an enormous amount of data that is publicly available for anyone to use and study. Also, just to give a few examples, the algorithms programmed allow us to obtain information such as the frequency of tweets over a period of time, the number of tweets for each sentiment degree and the geolocation of positive/negative tweets. This information

would be useful to any person or business with a big presence in social media to get instant feedback during political campaigns or product launches.

Economic analysis and cost evaluation

For the economic analysis and cost evaluation we will be taking a look at all the resources used during the project and how much would they effect the final cost of the project. This analysis will be structured in the two following categories:

- **Personnel costs.** In this category, the total of hours destined into the developing of the project will be transformed into a monetary cost. As an approximate cost per hour I will be using the relation: **9 €/h.**

The Following graph shows the number of Hours dedicated into the development of the project.

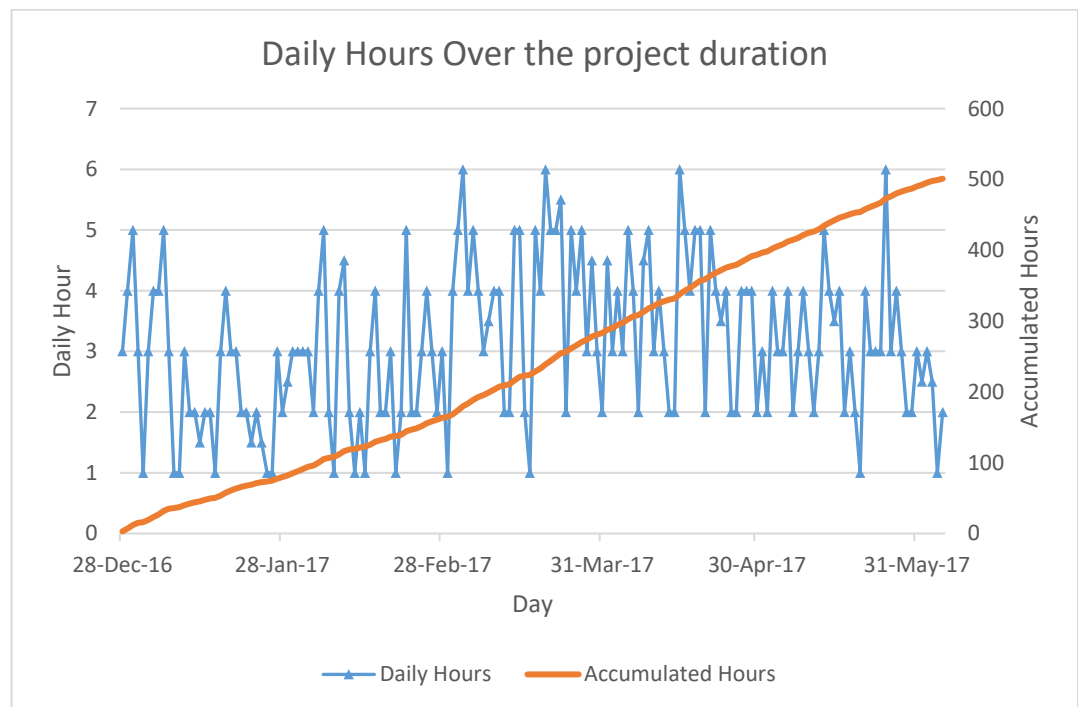


Figure 10.1. Daily hours of work over the duration of the project.

However, this graph only shows the hours recorded for each day since I started on working with the project (28th December 2016), and until the project is delivered. I also estimated a total of 60 hours for other tasks that could be related to the project, as well as a 10h dedication into the development/preparation of the presentation on the last week of June.

The following pie chart shows the distribution of time for each of the project tasks:

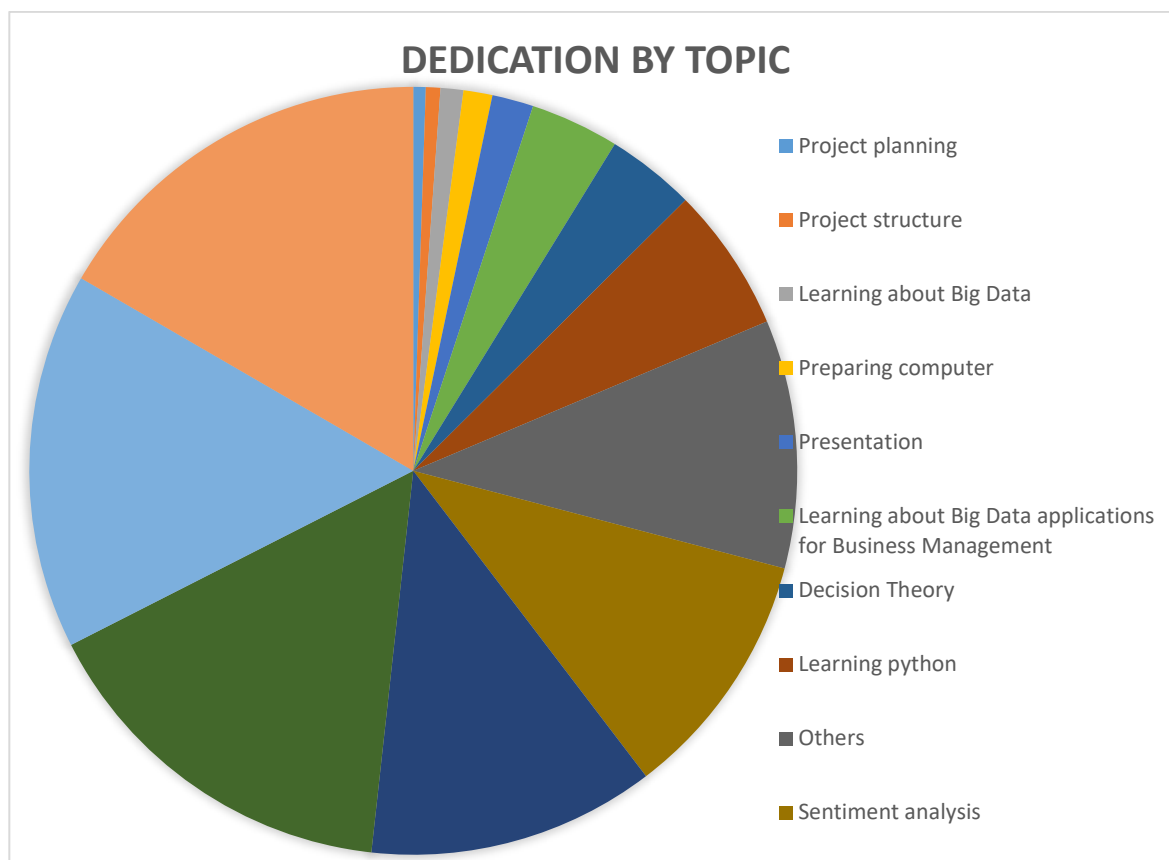


Figure 10.2. Dedication by topic.

Given all that has been previously explained, the following Table contains the values for each topic as well as the total number of hours dedicated into the project:

Topic	Hours
Project planning	3
Project structure	3.5
Learning about Big Data	5.5
Preparing computer	7
Presentation	10
Learning about Big Data applications for Business Management	21.5

Decision Theory	21.5
Learning python	35
Others	60
Sentiment analysis	60.5
Business Intelligence	69.5
Twitter sentiment analysis	90.5
Business Intelligence Applications	91
Writing the rest of the project	95.5
TOTAL	574

Table 10.1. Number of hours dedicated by topic.

By applying the cost of **9 €/h** the whole personnel costs would be: **5166 €**.

- **Software.** During the development of the project, the software used is only conformed by a set of Open-source tools or software with free versions available. Therefore, the total cost for software used is **0 €**.

Bibliography

- [1] Business Dictionary, “Decision Making Definition.”
- [2] S. Ove Hansson, *Decision theory: A Brief Introduction*. Stockholm: Royal Institute of Technology, 1994.
- [3] C. Vercellis, *Business Intelligence*. John Wiley & Sons, Ltd., 2009.
- [4] Business Dictionary, “Business Intelligence Definition.” [Online]. Available: <http://www.businessdictionary.com/definition/data-warehouse.html>.
- [5] I. Chen and K. Popovich, “Understanding customer relationship management (CRM),” *Bus. Process Manag. J.*, vol. 9, no. 5, pp. 672–688, 2003.
- [6] A. J. Bush, J. B. Moore, and R. Rocco, “Understanding sales force automation outcomes: A managerial perspective,” *Ind. Mark. Manag.*, vol. 34, no. 4 SPEC ISS., pp. 369–377, 2005.
- [7] S. Li, B. Ragu-Nathan, T. S. Ragu-Nathan, and S. Subba Rao, “The impact of supply chain management practices on competitive advantage and organizational performance,” *Omega*, vol. 34, no. 2, pp. 107–124, 2006.
- [8] A. Gunasekaran, C. Patel, and R. E. McGaughey, “A framework for supply chain performance measurement,” *Int. J. Prod. Econ.*, vol. 87, no. 3, pp. 333–347, 2004.
- [9] B. Liu, *Sentiment Analysis and Opinion Mining*, no. May. Morgan & Claypool Publishers, 2012.
- [10] C. J. Hutto and E. Gilbert, “Vader: A parsimonious rule-based model for sentiment analysis of social media text,” *Eighth Int. AAAI Conf. Weblogs ...*, pp. 216–225, 2014.

Annex A. Application Code

A1. Data Capture

```

from tweepy.streaming import StreamListener
from tweepy import OAuthHandler
from tweepy import Stream
import json

access_token = "248654713-z4gpU09YGKU93DdoEfK8oFDnWcMQk1iJgrqYJtGM"
access_token_secret = "5DMtDiGQXeuKcFFTpsMBt9DEpOB6cZPsKpHmEddIgkRvN"
consumer_key = "wCM46nj7KcdTIKObaM8nVh68V"
consumer_secret = "lCa9fIG4Cp7jzOt9EPLqEoxpcODzIJq45vebOyZaWbFB2VtUwb"

class StdOutListener(StreamListener):
    def on_data(self, data):
        try:
            with open('data.json', 'a') as f:
                f.write(data)
            all_data=json.loads(data)
            return True
        except KeyError:
            return True
        except IncompleteRead:
            return true

    def on_error(self, status):
        print status
        return

if __name__ == '__main__':

    l = StdOutListener()
    auth = OAuthHandler(consumer_key, consumer_secret)
    auth.set_access_token(access_token, access_token_secret)
    stream = Stream(auth, l)

    stream.filter(track=['Trump' ], languages=["en"], async=False)

```

A2. Added accumulative sentiment

```
from nltk.sentiment.vader import SentimentIntensityAnalyzer
import json
import matplotlib.pyplot as plt
from matplotlib import style

style.use("ggplot")

sentimentlist = []
sentimenttime = []
a=0
sid = SentimentIntensityAnalyzer()

with open ('data.json', 'r') as f:
    for line in f:
        try:
            tweet=json.loads(line)
            text=tweet['text'].encode("utf-8")
            if ('@RT' and 'RT @') not in text:
                ss = sid.polarity_scores(text)
                sentimentlist.append(ss['compound'])
                a+=float(ss['compound'])
                sentimenttime.append(a)
        except:
            continue

plt.title('Added accumulative sentiment')
plt.xlabel('Tweet Number')
plt.ylabel('Sentiment')
plt.plot(sentimenttime)
plt.show()
```

A3. Accumulative average sentiment

```

from nltk.sentiment.vader import SentimentIntensityAnalyzer
import json
import matplotlib.pyplot as plt
from matplotlib import style

style.use("ggplot")

sentimentlist = []
sentimenttime = []
a=0
b=0
sid = SentimentIntensityAnalyzer()

with open ('data.json', 'r') as f:
    for line in f:
        try:
            tweet=json.loads(line)
            text=tweet['text'].encode("utf-8")
            if ('@RT' and 'RT @') not in text:
                ss = sid.polarity_scores(text)
                sentimentlist.append(ss['compound'])
                a+=float(ss['compound'])
                b+=1
                sentimenttime.append(a/b)
        except:
            continue

plt.title('Accumulative average sentiment')
plt.xlabel('Tweet Number')
plt.ylabel('Sentiment Average')
plt.plot(sentimenttime)
plt.show()

```

A4. Sentiment degree frequency

```
from nltk.sentiment.vader import SentimentIntensityAnalyzer
import json
import matplotlib.pyplot as plt
from matplotlib import style
from collections import Counter
from operator import itemgetter
import collections

sentimentlist = []
sid = SentimentIntensityAnalyzer()

with open ('data.json', 'r') as f:
    for line in f:
        try:
            tweet=json.loads(line)
            text=tweet['text'].encode("utf-8")
            if ('@RT' and 'RT @') not in text:
                ss = sid.polarity_scores(text)
                sentimentlist.append(round(ss['compound'],1))
        except:
            continue

sentiment_freq = Counter(sentimentlist)
lt_ordered_data=collections.OrderedDict(sorted(sentiment_freq.items()))

plt.title('Sentiment Degree Frequency')
plt.xlabel('Sentiment Degree')
plt.ylabel('Number of tweets')
plt.bar(range(len(lt_ordered_data)), lt_ordered_data.values(), align='center')
plt.xticks(range(len(lt_ordered_data)), list(lt_ordered_data.keys()), rotation='vertical')
plt.show()
```

A5. Timely frequency

```
import json
import numpy as np
import matplotlib.pyplot as plt
from matplotlib import style
from collections import Counter
from operator import itemgetter
import collections
from datetime import datetime
from dateutil import parser

timelist = []

with open('data.json', 'r') as f:
    for line in f:
        try:
            tweet=json.loads(line)
            text=tweet['created_at'].encode("utf-8")
            if ('@RT' and 'RT @') not in text:
                time=tweet['created_at']
                dt=parser.parse(time)
                dtwithoutseconds = dt.replace(second=0, microsecond=0)
                dt2=dtwithoutseconds.time().isoformat()
                timelist.append(dt2)
        except:
            continue

time_freq = Counter(timelist)
lt_ordered_data=collections.OrderedDict(sorted(time_freq.items()))

plt.title('Timely Frequency')
plt.xlabel('Time')
plt.ylabel('Number of tweets')
plt.bar(range(len(lt_ordered_data)), lt_ordered_data.values(), align='center')
plt.xticks(range(len(lt_ordered_data)), list(lt_ordered_data.keys()), rotation='vertical')
plt.show()
```


A6. Number of positive and negative tweets

```
import json
import numpy as np
import matplotlib.pyplot as plt
from matplotlib import style
from collections import Counter
from operator import itemgetter
import collections
from datetime import datetime
from dateutil import parser

timelist = []

with open('data.json', 'r') as f:
    for line in f:
        try:
            tweet=json.loads(line)
            text=tweet['created_at'].encode("utf-8")
            if ('@RT' and 'RT @') not in text:
                time=tweet['created_at']
                dt=parser.parse(time)
                dtwithoutseconds = dt.replace(second=0, microsecond=0)
                dt2=dtwithoutseconds.time().isoformat()
                timelist.append(dt2)
        except:
            continue

time_freq = Counter(timelist)
lt_ordered_data=collections.OrderedDict(sorted(time_freq.items()))

plt.title('Timely Frequency')
plt.xlabel('Time')
plt.ylabel('Number of tweets')
plt.bar(range(len(lt_ordered_data)), lt_ordered_data.values(), align='center')
plt.xticks(range(len(lt_ordered_data)), list(lt_ordered_data.keys()), rotation='vertical')
plt.show()
```

A7. Number of tweets by country

```

from nltk.sentiment.vader import SentimentIntensityAnalyzer
import json
import matplotlib.pyplot as plt
from matplotlib import style
from collections import Counter
from operator import itemgetter
import collections

countrylist = []

with open ('data.json', 'r') as f:
    for line in f:
        try:
            tweet=json.loads(line)
            text=tweet['text'].encode("utf-8")
            if ('RT' and '@RT') not in text:
                countrylist.append(tweet['place']['country_code'])
        except:
            continue

country_freq = Counter(countrylist)
lt_ordered_data=collections.OrderedDict(sorted(country_freq.items()))

plt.title('Number of tweets by Country')
plt.xlabel('Country')
plt.ylabel('Number of tweets')
plt.bar(range(len(lt_ordered_data)), lt_ordered_data.values(), align='center')
plt.xticks(range(len(lt_ordered_data)), list(lt_ordered_data.keys()), rotation='vertical')
plt.show()

```

A8. Number of positive/negative tweets by country

```
from nltk.sentiment.vader import SentimentIntensityAnalyzer
import json
import matplotlib.pyplot as plt
from matplotlib import style
from collections import Counter
from operator import itemgetter
import collections

pos_location=[]
neg_location=[]
sid = SentimentIntensityAnalyzer()

with open ('data.json', 'r') as f:
    for line in f:
        try:
            tweet=json.loads(line)
            text=tweet['text'].encode("utf-8")
            if ('RT' and '@RT') not in text:
                ss = sid.polarity_scores(text)
                if ss['compound']>0:
                    pos_location.append(tweet['place']['country_code'])
        except:
            continue

pos_freq = Counter(pos_location)
pos_ordered_data=collections.OrderedDict(sorted(pos_freq.items()))
plt.title('Number of positive tweets by Country')
plt.xlabel('Country')
plt.ylabel('Number of tweets')
plt.bar(range(len(pos_ordered_data)), pos_ordered_data.values(), align='center')
plt.xticks(range(len(pos_ordered_data)), list(pos_ordered_data.keys()), rotation='vertical')
plt.show()
```

A9. Geolocation of positive/negative tweets

```

from nltk.sentiment.vader import SentimentIntensityAnalyzer
import json
import matplotlib.pyplot as plt
from matplotlib import style
from collections import Counter
from operator import itemgetter
import collections
import numpy as np
import matplotlib.patches as mpatches
import gmaps
import gmaps.datasets

pos_location=[]
neg_location=[]
sid = SentimentIntensityAnalyzer()
gmaps.configure(api_key="AlzaSyCymMgSJnFiRyaNpipg4BYe-QBFvwjq2B4")

with open ('data.json', 'r') as f:
    for line in f:
        try:
            tweet=json.loads(line)
            text=tweet['text'].encode("utf-8")
            if ('@RT' and 'RT') not in text:
                if 'null' not in tweet['place']['bounding_box']['coordinates']:
                    ss = sid.polarity_scores(text)
                    lat=tweet['place']['bounding_box']['coordinates'][0][0][1]
                    lon=tweet['place']['bounding_box']['coordinates'][0][0][0]
                    if ss['compound']<0:
                        neg_location.append((lat, lon))
                    if ss['compound']>0:
                        pos_location.append((lat, lon))
        except:
            continue

pos_layer = gmaps.symbol_layer(
    pos_location, fill_color="green", stroke_color="green", scale=2)

neg_layer = gmaps.symbol_layer(
    neg_location, fill_color="red", stroke_color="red", scale=2)

m = gmaps.Map()
m.add_layer(pos_layer)
m.add_layer(neg_layer)
m

```

